ISSN: 0975-766X
CODEN: IJPTFI
Available Online through *Research Article*
www.ijptonline.com

K-NN SEARCH ALGORITHM AND EHSD FOR ENHANCED DATA PRIVACY IN CLOUD

V. Rajalakshmi*
Assistant Professor, Department of Information Technology, Sathyabama University, Jeppiaar Nagar, Chennai-119. India.
Email: rajalakshmi.it@sathyabamauniversity.ac.in

Received on 04-03-2016 *Accepted on 25-03-2016*

Abstract

Cloud computing infrastructures are popular. By using cloud users can save their cost but some of the data owners are undecided to put their data's in cloud because, sometimes the data may be hack or attack. The data at a variety of sensitivity levels (e.g., in medicine), valuable (e.g., in astronomy) or otherwise confidential without compromising security. This paper focus on secure query processing by the data is to be revealed only to trusted client, not to the service provider. Client who queries the server for the most similar data objects. Outsourcing offers data to the data owner. The paper presents techniques that transform the data prior to supplying it to the service provider for similarity queries on the transformed data. They are then further extended to offer an intuitive privacy guarantee. Empirical studies with data demonstrate that the techniques are capable of offering privacy while enabling efficient and accurate processing of similarity queries.

Keywords: Outsourced data, EHSD, KHN search algorithm, Data Privacy, Query processing.

Introduction

In the outsourced database environment, a data owner access from outsource his/her database to a service provider, in order to reduce cost for data storage and management. Only both authorized users and a data owner are allowed to approach the outsourced data, but not the third parties. The traditional spatial

databases owners want to outsource their databases to the service provider so that they can manage the spatial data efficiently. In this context, the issue of privacy preservation is very important in spatial database outsourcing because a user's location data is valuable and sensitive against unauthorized accesses. Security issues have been classified into sensitive data access, data segregation, privacy, bug exploitation, recovery, accountability, malicious insiders, management console security, account control, and multi-tenancy issues. Solutions to different cloud security issues vary, from cryptography, particularly public key infrastructure (PKI), to use of multiple cloud providers, standardization of APIs. Advances in digital measurement and engineering technologies enable the capture of massive amounts of data in fields such as astronomy, medicine, and seismology. The effort for data collection and processing, then its potential utility for research or business, creates value for the data owner. He wishes to store and access by himself, colleagues, and other (trusted) scientists or customers. This can be supported by outsourced servers that offer low storage costs for large databases. For instance, outsourcing based on cloud computing is becoming increasingly effective, as it promises pay-as-you go, low storage costs as well as easy data access. However, care needs to be taken to protect data that is valuable or sensitive against unauthorized access. In this context, call any item in a data collection an object, individuals with authorized access query users, and the entity offering the storage service. The ultimate aim of the project is to improve query efficiency and also provide privacy (data security).

Related Works:

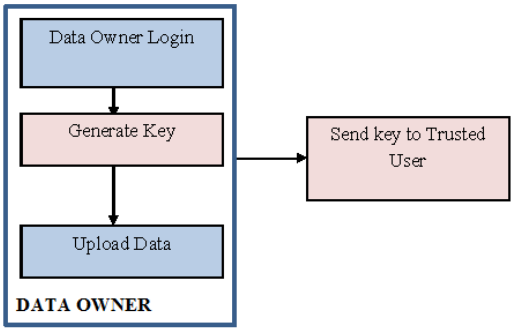
The objective of our work is to improve query efficiency in immense dataset. But in Existing solutions either offer query efficiency at no privacy. Many applications in science and business rely on similarity search of metric data other than time series and vector data. Computer-aided gene sequencing uses the similarity between an unknown sequence from one species and a known sequence from a closely related species to predict the former's function, which can be represented as a labelled graph. The valuable data in a metric space are searched based on a similarity measure. When this data is outsourced, they must be

secured against leaks or attacks. Using RASP [1] data perturbation method to provide efficient range query and kNN query services but not for data privacy. The RASP method will secure the multidimensional range and increase the working process of query [2] RASP method will use the four concepts of the CPEL criteria, but increase the complexity of query service. Edit distance is widely used for ensuring the similarity between two strings. B+tree [3] based approach is proposed to answer edit distance based string similarity queries, using pruning techniques employed in the metric space. First, split the string collection into partitions and index strings in all partitions using a single B+-tree based on the distances of these strings to their corresponding reference strings. Finally, propose two approaches to efficiently answer range and KNN queries but it is unnecessary KNN algorithm is quite enough to find the distance of related object. A privacy- preserving query processing algorithm which performs on encrypted spatial database by retrieving k-nearest neighbour (k-NN)[4],but not capable for data partitions. Propose RT- CAN method which is a multi-dimensional indexing scheme in epic. RT-CAN [5] integrates CAN- based routing protocol and the R-tree based indexing scheme to support efficient multi-dimensional query processing in a Cloud system. Maintaining a global multi-dimensional search index, making the scheme scalable in terms of data volume. But they not looking for data privacy and secure Query processing. [6] Proposed Query scheduling algorithm which Least loaded processor first. Prevent from bottlenecks at the query receptionist side, it has a drawback that running cost is high. M. L. Yiu, et al., [9] proposed the distance-oriented transformation techniques called MPT (Metric preservation technique). This method converts an original spatial database in a metric space into another metric space datasets, but using it for evaluating the NN query. Then using Digital watermarking [10] establish the data owner's identity on the data. Additional information stored in the data helps prove ownership, but it cannot prevent an attacker from illegally copying the data set. It encrypts both distances and anchor information by using an order preserving encryption scheme (OPES [12]). Using these various different concepts, concentrate only on the query efficiency and also provide privacy (data security).

System Design

The Proposed System to Security and query efficiency is challenging issues in cloud computing, these issues motivated to propose the Enhanced HSD method in cloud computing environment, to provide data privacy**.** Three other transformation methods (EHI, MPT, FDH) they capture various trade-offs among data privacy and query cost and accuracy are combined. A similarity search techniques for sensitive metric data, e.g., bioinformatics data that enable outsourcing of such search. To improve query efficiency use dataset are indexed and to provide privacy to metric dataset using Enhanced HSD method. Various advantages of this method are

- privacy guarantee
- Security
- Approximation of the query result
- Encrypted index-based technique.
- Low storage costs for large databases.



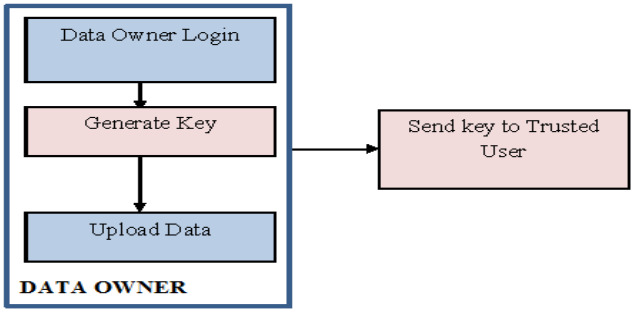
[Figure 1 Outsourcing Data].

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 (e.g., actual time series, graphs, strings), and a key to be used for transformation. First, the data owner applies a transformation function with a EHSD key to convert P into a set P0 of transformed objects, and uploads the set P0 to the server (see step A1 in the figure 1). The server builds an index structure by EHI 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).

Encrypted Hierarchical Index Search (EHI)

This section for performing NN search on an encrypted hierarchical index stored at the server. This method offers perfect data privacy for the data owner, but it incurs multiple communication round trips during query processing. Since the tree index stored at the server is encrypted, the server cannot process the NN query by itself. The communication between the client and the server needs to be developed in order to answer the NN query correctly.



[Figure 2. Encrypted Hierarchical Index Search (EHI)]

Metric Preserving Transformation (MPT)

In this section, a method is developed using metric preserving transformation (MPT), for evaluating the NN query. Unlike the EHI method, MPT incurs only 2 rounds of communication during the query phase. The basic idea behind MPT is to pick a small subset of the data set P as the set of anchor objects and then assign each object of P to its nearest anchor.

Flexible Distance-Based Hashing (FDH)

In this section, a hashing-based technique, called flexible distance-based hashing, for processing the NN query. The main advantage of this technique is that the server always returns a constant-sized candidate set. 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

Algorithm

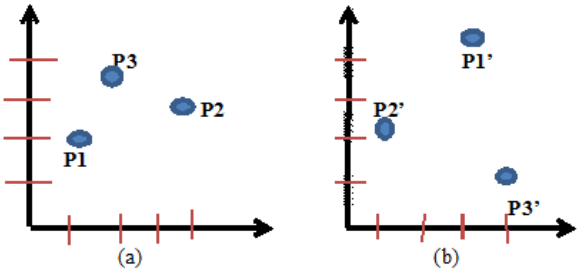
Enhanced HSD Method: The Hierarchical Space Division (HSD), Error-Based Transformation (ERB) and [8] HSD* benefits are combined. HSD is strong against the attack done by adversaries who have the subset of database information. The ERB shows bounded errors into the data that are reversible with the help of the secure hash function. The HSD transformation technique partitions the space and then applies a distinct linear transformation function for each partition. However, an attacker who knows two points in the same partition can infer the precise transformation function used in that partition. Although the ERB transformation is resistant against the above attack, it incurs high query costs for typical values. A novel transformation called Enhanced EHSD (HSD*) that exploits the benefits of HSD and ERB: By this EHSD applies the HSD transformation for the global space and then performs the ERB transformation at the level of partitions. The advantages include

- A single round of communication.
- Efficient query processing,
- Robustness against the general attack, and
- Robustness against the tailored attack when the adversary has polynomial bounded Computational Power.

```
Alogrithm1 EHSD Transformation Key(Point P, Transformed point S, Level M,
Transformspace (x1',xr')X(y1',yr'))
1: H,a Root Node H is created;
2: if M mod 2=0 then    /*M is an even number
3:          Adding 2 value to the H Node;
            H.i:= the P's Median value of X coordinates;
            H.i':= the S's Median value of X coordinates;
4:          if M>0 then
5:          P is Splitting into PL and PR such that
                PR:={ p ∈ P with H.i } ;
                PL:={ p ∈ P };
6:          S is Splitting into SL and SR such that
                SR:={ p ∈ P with H.i' } ;
                SL:={ p ∈ P };
7:          Set HR and H L as the right and left child node of H respectively;
              HR :=EHSD_Key(M-1, PR, SR, (H.i',xr')X(y1',yr') );
              HL :=EHSD_Key(M-1, PL, SL, (x1', H.i')X(y1',yr') );
8: else
            Steps 3-7 for Y coordinates  for points P and S;
9:return H;
```

[Figure 3 EHSD Transformation Key Generation]

Algorithm 1 shows the HSD pseudo-code for extracting key parameters from P and S. Initially the transformed space, ln Line 1 a root node H is created. If M is an even number, only the X coordinate values are processed in steps 3–7 else the Y coordinate values are processeded in steps 8. In step 3, adding two values for node H the following values: (i) H.i, the median X value of the points in P, and (ii) H.i', the median X value of the points in S. When M is non-zero in step 4, the set P is splitted into two sets, such that PK contains all tuples of P with X values above H.i and PL has the remaining tuples of P. Similarly, the set S is divided into two sets SK and SL. The algorithm is first performed on PL and SL repeatedly to obtain the left child node (i.e., HL) of the root node H. Then, the algorithm is applied on PK and SK repeatedly to obtain the right child node (i.e., HR) of H. Finally, the current node H is returned to the caller. The output of Algorithm 1 is an M-level tree with 2M – 1 Nodes, where each node H stores the splitting X-values (or Y -values) H.i and H.i' for the original.



[Figure 4 EHSD key Generated to Transfer the Data points]

Figure 4 shows Data point transformation. The transformation key of EHSD contains 2(2M − 1) parameters, where M is an even. The original point set (a) from the data owner is denoted by P. A target point set P' (b) is used to capture the data distribution in the transformed space to mislead the attacker by altering the distribution of the points in the transformed space. Analysis of attacks, now examine the tailored attack on the EHSD method. As in HSD, the attacker who calculate a system of equations by substituting the coordinates of known points. However, even if the attacker knows two points in the same partition, the SHA values in the transformed points protect the leakage of the transformation key. EHSD inherits its security from ERB and thus computationally hard to break.

K-Nearest Neighbor Query Search

A method is developed for processing k nearest neighbor (kNN) queries on transformed data. It is designed to the retrieval of exact kNN results from the database. Also, the proposed method is generic and suitable for all the transformations (HSD, ERB, HSD*) described earlier. K denotes positive integer and this query are used to find the value of nearest neighbour point to k. Queries are constructed by using structured query language. Query services are the method for services that are exposed through an implementation of service provider. kNN query in cloud provide secure, fast storing and retrieving process of encryption and decryption of a data from database.

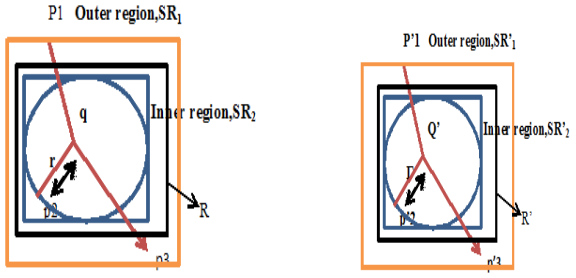
K-NN Query meets the following requirements:

1) It enables high query accuracy.
2) It enables efficient query processing in terms of communication cost.
3) It supports the insertion and deletion of data objects.

```
Transformed Key (Query point q,Value k, Point P, Transformation TR, Key K)
1: Set Result set,RS :=(p,dist(q,p));
2: Assign k values;
3: Set ? := highest distance of q;   /*top distance of RS
4: q'':= apply transformation on q;
5: Using INN at q'' to the server;
6: repeat
7:        retrieval the Next  p'' closest to q'' from server;
8:        Set Γ:= the distance of  (p'',q'');
9:        Set Square Region SR'' := q'' as center and side length 2Γ;
10:       p=apply inverse transformation on p'';
11:       if dist(p,q)<?  then
              Update RS and ?  by p;
              Set square region R:=q as center and side length 2? ;
              Set R'' := by applying transformation on R;
12: Until SR'' covers R'';
13: INN at server is terminated;
14:RS is taken as result to return;
```

[Figure 5 KNN Search Method at Client Side]

Algorithm 2 shows the pseudo-code of a client-side method at the server for performing kNN search on transformed data. The querying user specifies the query point q and the number for k value. In addition to this, the algorithm also needs to know the transformation method TR and the key value K used for the transformation. First, it consider a result heap RS for maintaining the k points closest to q seen so far. The variable γ denotes the top distance stored in RS. Query point q is converted to q' in the transformed space. The client use Incremental Nearest Neighbor search (INN) query to the server, in order to retrieve (transformed) points in ascending order of their distances from q. It is worth noticing that replace the distance metric used in INN by the L∞ norm. In step 7, the point p' is retrieved from the server as the next closest point to q' (in the transformed space). The variable τ represents the largest L∞ distance from q' to P' seen. A square region SQ with q as its center and 2τ as its side length are defined. The INN search guarantees that any (transformed) point in SQ has been retrieved. Client decodes the transformed point P' back to its original point p. In case q is closer to p than some existing point in the result heap RS, update RS and the kNN distance γ by using p.Calculate another square region R with q as its center and 2γ as its side length. Observe the region R is to cover the actual kNN of q, regardless of whether the actual kNN results have been retrieved or not. In step 11, apply the transformation to convert R into R'. The loop in Step 6–12 terminates if the searched region SQ covers each rectangle of R', as the actual kNN of q must been retrieved. After that, the client terminates the incremental NN query at the server and sending the points of RS as the result to the query user.



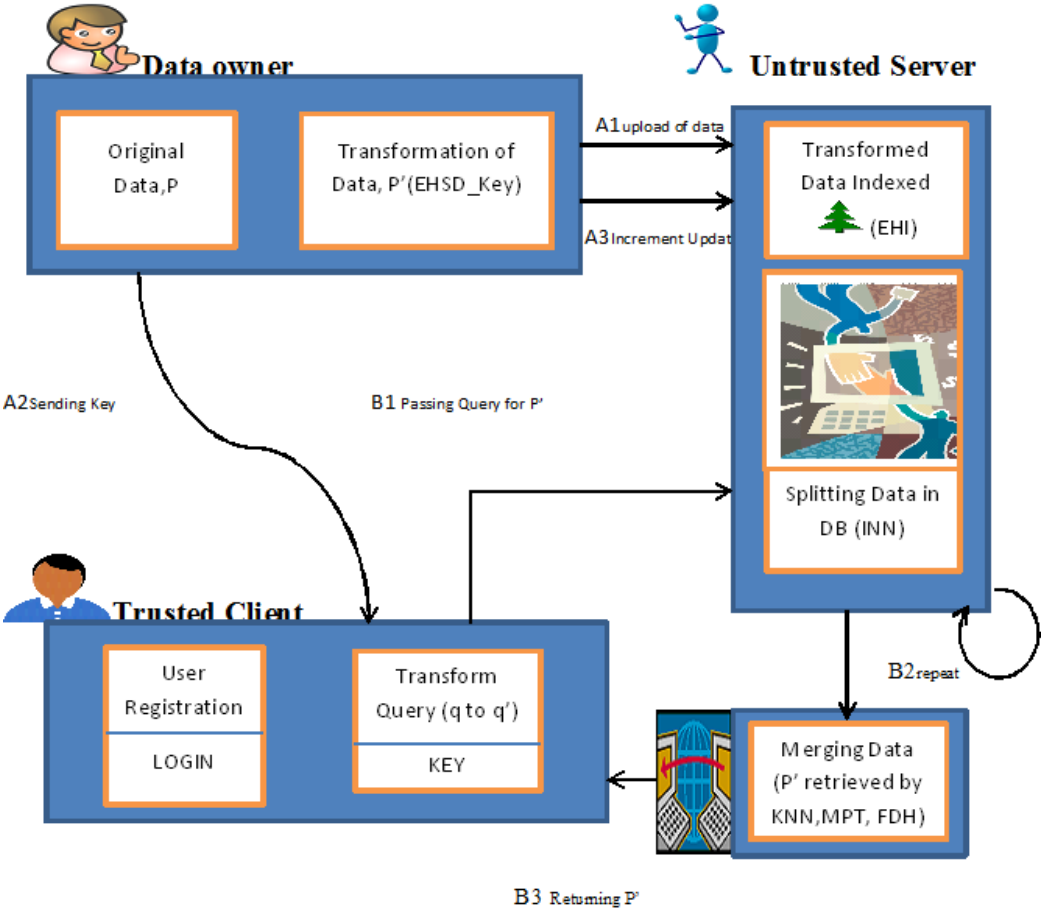
(a) (b)

[Figure 6(a) Example of transformed incremental kNN search, at k = 1 a Transformed space]
[Figure 6(b) original space]

Implementation

The method is proposed for cloud computing using the algorithms to provide good accuracy and data privacy. In this proposed method HSD, MTP, FDH to provide privacy and a single round of communication are used.

In Data Collection, All the required data's are stored in database. Data is collected are user defined. These data are given as input to the data owner. Data Privacy is done by hiding the data in encrypted form by using symmetric encrypted algorithm while client want to see data he has to decrypt the data using key which will be given to him then he can able to see his data. Accuracy is provided by using HSD Algorithm which partitions the data in the server and while client wants their data they will merge and shows the data in the encrypted manner.



[Figure 7: System Architecture]

It consists of three entities: a data owner, a trusted query user, and an untrusted server. One hand, the data owner wishes to upload 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 (e.g., actual time series, graphs, strings), and a key to be used for transformation.

First, the data owner applies a transformation function with a EHSD key to convert P into a set P0 of transformed objects, and uploads the set P0 to the server (see step A1 in the figure). The server builds an index EHI structure on the set P0 in order to facilitate efficient search. In addition, the data owner applies a standard Splitting method (e.g. AES) on the set of original objects; the resulting split objects (with their IDs) are uploaded to the server and stored in a relational table.

Next, the data owner informs every user of the transformation key (step A2). In future, the data owner can perform incremental insertion/deletion of objects. (step A3). At query time, a trusted user applies the transformation function (with a key) to the query q and then sends the transformed query q' to the server (step B1).

Then, the server processes the query (see step B2), and reports the results back to the user (step B3). The user merges the retrieved results back into the actual results. Observe that these results contain only the IDs of the actual objects.

Experimental Evaluation

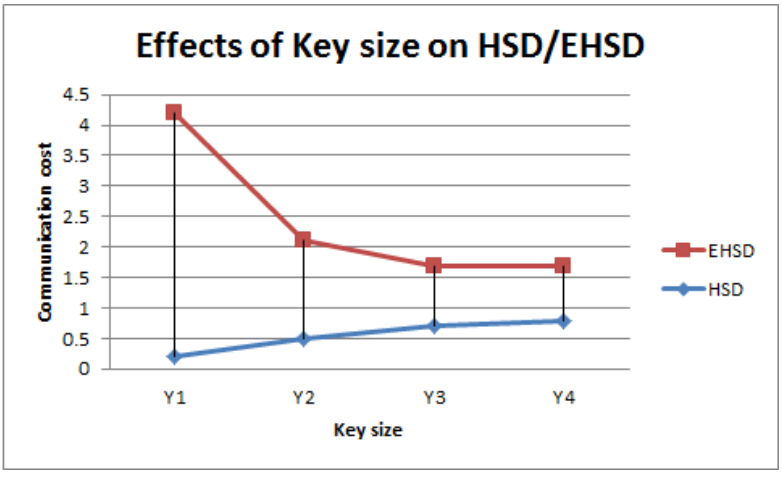
The transformation methods HSD, ERB, EHSD for query processing cost and a bulk of data sets used. For fair comparison, the existing MPT, FDH method to support k-NN query processing in transformation space are extended. For the performance of k-NN query processing, the number of key size ϒ from 64 to 4096 with HSD and EHSD to estimate communication cost. Both algorithms are implemented in Visual studio 2008 on Window 7 professional SQL 2005 operating system with Intel Core(TM) i3 CPU 530 @ 2.93GHz and 2GB RAM. Every result reported in this paper is the average value of 100 k-NN query processing.

[Table 1 KNN Query Cost for different data sets].

KNN Query cost				
DATASET	D1	D2	D3	D4
BULK	119.23	356.69	2413.92	3433.84
HSD	0.64	1.69	0.6	0.37
ERB	8.74	47.19	186.44	254.65
EHSD	0.73	1.91	1.06	0.9

[Table 2 Communication Cost for different data sets].

COMMUNICATION COST				
KeySeize	G1	G2	G3	G4
HSD	0.2	0.5	0.7	0.8
EHSD	4	1.6	1	0.9



[Figure 8 Effects of Key size on HSD/EHSD]

Conclusion

To improve query efficiency and to provide privacy to metric dataset the Enhanced HSD method is developed. For performing NN search on an encrypted hierarchical index stored at the server. This method offers perfect data privacy for the data owner. Existing solutions either offer query efficiency at no privacy, or they offer complete data privacy while sacrificing query efficiency. To overcome these issue Enhanced HSD method and Encrypted Hierarchical Index Search which provide interesting trade-offs between query cost and accuracy. They are then further extended to offer an intuitive privacy guarantee. This paper discussed issues in providing privacy and query processing.

References

1. Huiqi Xu, Shumin Guo, Keke Chen, "Building Confidential and Efficient Query Services in the Cloud with RASP Data Perturbation", IEEE Transaction on Knowledge and data Engineering, Vol. 26, No. 2, Feb 2014
2. E.Saral Elizabeth1, Ms.K. Padmaveni2, "Confidential and Efficient Query Services in the Cloud", IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 2, Issue 1, Feb-Mar, 2014 -ISSN: 2320 – 8791
3. Wei Lu, Xiaoyong Du, Marios Hadjieleftheriou, Beng Chin Ooi, "Efficiently Supporting Edit Distance based String Similarity Search Using B+-trees", 2012 IEEE Computer Society Kanai.
4. S. Papadopoulos, S. Bakiras, and D. Papadias, "Nearest Neighbor Search with Strong Location Privacy," Proc. Very Large Databases Conf. (VLDB), 2010.
5. Jinbao Wang , Sai Wu , Hong Gao , Jianzhong Li , Beng Chin Ooi, "Indexing Multi-dimensional Data in a Cloud System" , SIGMOD'10, June 6–11, 2010 ACM 978-1-4503-0032-2/10/06.
6. Veronica Gil-Costa and Mauricio Marin"Load Balancing Query Processing inMetric-Space Similarity Search."
7. D. Sacharidis, K. Mouratidis and D. Papadias, "k-Anonymity in the Presence of External Databases", IEEE Transactions on Knowledge and Data, vol. 22, no. 3, (2010).
8. M. L. Yiu, G. Ghinita, C. S. Jensen and P. Kalnis, "Outsourcing Search Services on Private Spatial Data", Proceeding of the 25th IEEE International Conference on Data Engineering, (2009) March 29-April 2, Shanghai, China.
9. M. L. Yiu, I.Assent, C. S. Jensen and P. Kalnis, "Outsourced Similarity Search on Metric Data Assets", IEEE Transactions on Knowledge and Data Engineering, vol. 24, no. 2, (2010).
10. R. Agrawal, P.J. Haas, and J. Kiernan, "Watermarking Relational Data: Framework, Algorithms and Analysis," The Int'l J. Very Large Data Bases, vol. 12, no. 2, pp. 157-169, 2003

11. X. Jiang, J. Gao, T. Wang and D. Yang, "Multiple sensitive association protection in the outsourced database", Proceedings of the 10th International Conference, Database Systems for Advanced Applications, (2010) April 1-4, Tsukuba, Japan.
12. R. Agrawal, J. Kiernan, R. Srikant and Y. Xu, "Order-Preserving Encryption for Numeric Data", Proceedings of ACM Special Interest Group on Management Of Data, (2004) June 13-18, Paris, France.
13. Man Lung Yiu, Gabriel Ghinita , Christian S, Jensen , Panos Kalnis, "Enabling search services on outsourced private spatial data ",The VLDB Journal (2010) 19:363–384.
14. Miyoung Jang, Min Yoon and Jae-Woo Chang, "A k-Nearest Neighbor Search Algorithm for Enhancing Data Privacy in Outsourced Spatial Databases", International Journal of Smart Home-Vol. 7, No. 3, May, 2013.

Corresponding Author:

V. Rajalakshmi*,

Email: *rajalakshmi.it@sathyabamauniversity.ac.in*