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.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Provide Integrity for Spatial Data Outsourcing

SivaSakthi S[1], Bini Tofflin R[2], Roslinmary M[3]

M.Tech/IT, Dr.Sivanthi Aditanar college of Engineering/Tiruchendur, Tamil nadu, India[1, 2, 3]

ABSTRACT: The necessity of outsourcing spatial data has grown rapidly over the past few years because of the popularity of location-based services and the abundant usage of smart phones and GPS-enabled devices. In the Outsourced Database model, data owners outsource their data management needs to third-party service providers, such a service provider offers mechanisms for the client to access data by spatial query and verify the correctness and completeness of the query result. In this paper, we focus on both correct and completeness of the query of that we propose an efficient scheme, called VN-Integrity, which allows client to verify the query result. We use voronoi diagram for spatial indexing, which derive the neighbors information for query verification. We evaluated VN-Integrity based on real-world data sets using mobile devices (Google Droid smart phones with Android OS) as query clients. Compared to the current state-of-the-art approaches (i.e., methods based on Merkle Hash Trees), our experiments show that VN-Auth produces significantly smaller verification objects and is more computationally efficient, especially for queries with low selectivity.

KEYWORDS: Spatial database outsourcing, location-based service, service provider, voronoi diagram, spatial query

I. INTRODUCTION

The Outsourced Spatial Database model contains three types of entities they are 1) the Data Owner (DO), 2) the Service provider (SP) and 3) the users or client. In this paradigm the SP manage and maintain the database of the Data Owners (DOs). That the SH should responsible for indexing the data and retrieving query result to the user who send spatial query to the Service provider. The SH is not the real owner of the data, there arise some major problems. First, the SH might return dishonest results out of its own interests. Second, query result might be modified by attackers by adding one or more fake records. In this paper we propose a result Verification approach for spatial queries, called VN-completeness of the query result.

In the outsourced spatial database model the DOs uploads their data to the SP. Before uploading the data to SH, using its private key, the DO digitally signs their data by generating a number of signatures. Then, the DO sends the signatures and the data to the SP. The client generates the query and sends it to the SH. When the SH receives the query, it generate Verification Object (VD) that contain result set along with the corresponding authentication information.

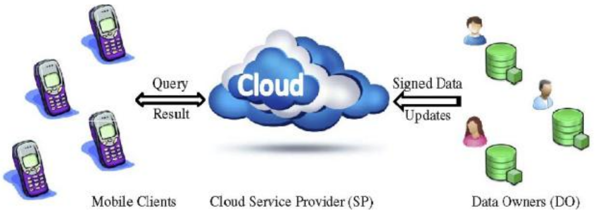


Fig. 1 System Architecture

The MRtree is essentially an R-tree deals with authentication information, i.e., hash digests. Leaf node of the MRtree stores a digest. The digest is computed on the concatenation of the binary representation of all objects in the node. Internal nodes are assigned a digest that summarizes the child nodes' minimum bounding rectangles (MBRs) and digests. Digests are computed in a bottom up fashion. The single digest at the root is signed by the DO. The resulting VO contains 1) all the objects in every leaf node visited and 2) the MBRs and digests of all the pruned nodes. Having this information, the client can reconstruct the root digest and compare it against the one that was signed by the owner. In addition, the client also examines the spatial relations between the query and each object/MBR included in the VO, in order to verify the correctness of the result.

Arguing about several drawbacks due to the structure of an MRtree and the verification process. First, during query processing more number of I/O access are made by embedding the information which is authenticated reduces the node fan out. Second, re-computation to be done from affected leaf node to the root, in the presence of update from the DO. If the updates are frequent performance of the query is degraded. Finally, the overhead of the VO can be significant, especially for queries that return only a few objects. This is due to the fact that the SP has to return all objects lying inside the leaf nodes that are visited during query processing. The result set includes only some object but the VO returns all object in the database.

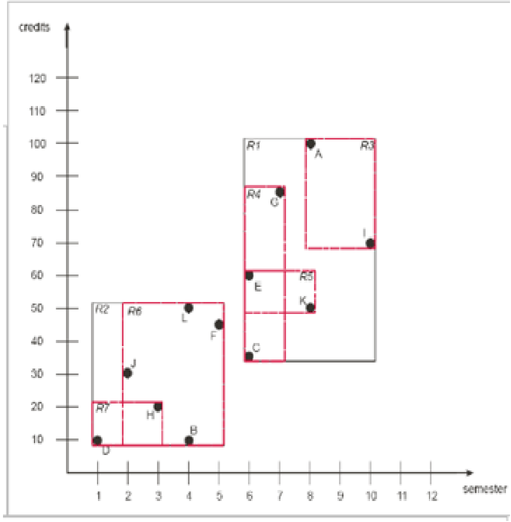


Fig.2 MBRs and Points

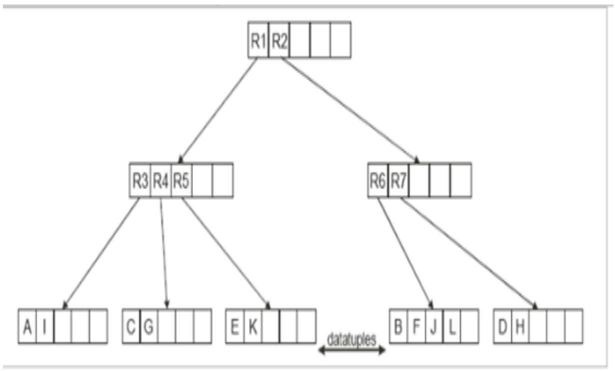


Fig.3 MR-tree

The VN-Integrity approach will authenticate the spatial query based on voronoi diagram from which the neighborhood information are derived. This approach separates the authentication information from the spatial index. Also in this approach the updates affect only the neighborhoods of the updated object. Here, for the result set of the KNN query and

Range query the VO returns the objects belong to the result set. The query verification process are take place in the client side, here android mobile is the client. The VN-Integrity produces significantly smaller verification objects and is more computationally efficient. The VN-Integrity approach handle more advanced spatial queries, such as reverse KNNs, K aggregate NNs, and spatial skyline query.

II. RELATED WORK

Many number of query authentication solutions have been proposed for auditing query result in outsourced relational databases. The MR-tree and MR*-tree are the space-efficient authentication data structures which are supporting for fast query processing and verification. The MR-tree computes hash digest by concatenating the entries in the tree node using its binary representation. To verify the correctness and completeness of the query result use VO. The VO includes 1) all visited objects and 2) MBRs and digests of all the pruned nodes. The MR-tree and MR*-tree never handle the data updates efficiently.

The Partially Materialized Digest scheme (PMD) [3] verifies one dimension queries and applies to both static and dynamic databases. This PMD approach apply separate the data indexing and the verification information, this reduce unnecessary costs when processing queries don't want the verification. For spatial queries two verification method are implement, one is Merkle R-tree and another one is the Partially Materialized KD-tree (PMKD). The Merkle R-tree support both the static and dynamic dataset. While the PMKD support the static dataset. Even though this approach reduce the transmission overhead between the service provider and the clients, it require changes to the DBMS software in order to support the embedded authentication information.

In the Query Integrity Assurance of Location-Based Services Accessing Outsourced Spatial Databases proposed a query integrity technique that does not require any modifications in the DBMS. This technique use replication copies of the data set for the verification purpose. The replicated copies are encrypted by different encryption key. By using this encryption method the client verify the correctness of the query result. However the attackers action or modifications may escape from the auditing process. But the VN-approach will efficient for query verification and it does not require any modifications in the DBMS software.

III. PRELIMINARIES

3.1 voronoi Diagrams

The Voronoi diagram (VD) is a data structure which is extremely efficient in exploring a local neighbourhood in a geometric space. The VD partitions space into cells whose interior points are closest to the generating point. The voronoi diagram of a given set $P = \{p_1, p_2,....., p_n\}$ of n points in R^d partitions the space of R^d into n regions. If p_m is a object in P , then the p_m should belongs to only one region and every point in that region is closer to p_m than to any other object of P in the Euclidean space. The region around p_m is called the voronoi cell of p_m. The voronoi diagram of P is the union of all voronoi cell. If two voronoi cell share the same voronoi edge then they are called the voronoi neighbours. The Delaunay triangulation is the dual graph which is construct by joining all the voronoi neighbours together.

Property 1) Given set of distinct points $P = \{p_1,p_2, ..., p_m\}$ belongs to R^m,the voronoi diagram and the corresponding Delaunay triangulation of P are unique.

Property 2) The average number of voronoi edges per voronoi polygon does not exceed six.

Property 3) Given the voronoi diagram of P, the nearest neighbour of query point q is p,if and only the query point q belongs to the voronoi cell of the point p.

Property 4) For the K (K>1) nearest neighbours in P to a query point q. Then, p_m is a voronoi neighbour of at least one point $p_r = \{p_1, p_2, \ldots, p_{m-1}\}$.

3.2 Signature Aggregation

The DO generates the hash digest by concatenate the binary representation of the object and its voronoi neighbors for computing one signature for each object in the dataset. Using this signature the client can verify the authentication of the query result. The RSA signatures are used for security purpose. When a client receives the VO from the SP, it verifies the aggregate signature using the public key of the DO. Specifically, the client simply performs a modular multiplication of the hash digests of all objects included in the VO, and verifies that the result matches the plaintext that is derived by decrypting the aggregate signature with the DO's public key. Note that, forging or altering a signature is computationally intractable for a polynomial time adversary. If the VO fails the signature verification process, the client considers the result as corrupted and the verification process terminates. Otherwise, it continues with the geometric verification.

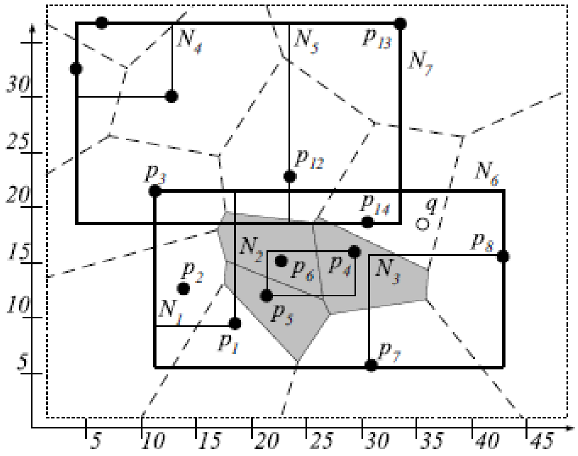


Fig. 6 Voronoi Diagram

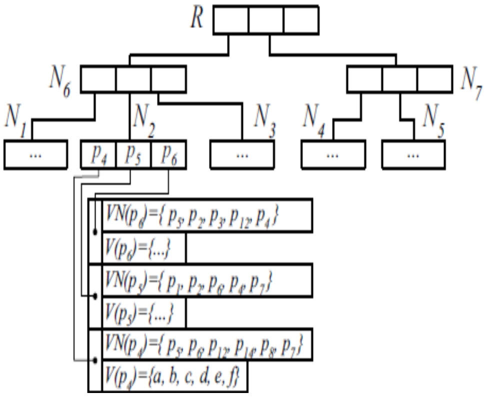

R
N6
N7
N1
N2
N3
N4
N5
...
p4
p5
p6
...
...
...
VN(p6)={p5, p2, p3, p12, p4}
V(p6)={...}
VN(p5)={p1, p2, p6, p4, p7}
V(p5)={...}
VN(p4)={p5, p6, p12, p14, p8, p7}
V(p4)={a, b, c, d, e, f}

Fig. 7 R-tree

IV. DATA TRANSFORMATION

The DO integrate a collection of n Point of Interest's (POIs) within a geographic region. In the dataset each POI has unique object. Example for POI i it's unique object will be p_i and it is in the form (p_i . location, p_i. tail). The location and tail are the attributes which store the spatial coordinates of the object and some additional information about the object respectively. The DO should attach neighbor and authentication information before transforming to the SP. The voronoi diagram is computed for the spatial data set by the DO to retrieves the voronoi neighbors of each POI. Also the DO add another attribute called neighbors attribute which stores the locations of all voronoi neighbor of each object. After finishing this above step's the DO to sign each individual object, so that the client can verify the authentication of the query result. The signature of the DO made on this information such as

$$S = \text{sign}(h(p_i.\text{location}|p_i.\text{tail}|p_i.\text{neighbors})),$$

Where h is a one-ay, collision-resistant hash function and "|" denotes the concatenation of two binary strings. Each object transformed object p_i at the DO's site has the form (p_i.location,p_i.tail,p_i.neighbors,p_i.S). After the completion of data transform process, the DO sent all objects to the SP. Upon receiving the database objects, the SP builds an appropriate spatial index and is then ready for query processing. Note that the leaf level of the index only stores pointers to the locations of the transformed objects on the disk.

V. AUTHENTICATION SPATIAL QUERIES

Many verification algorithms are introduced for typical location-based queries. And verify the authentication of the query result .

5.1 Query Processing at the SP

The query processing is held in the SP, instead of the Data Owner this due to the data outsourcing. The SP return the Verification Object(VO) as result set. The VO contains 1) a signature S that verifies the authentication of the query result 2) the result set with additional information which is useful for the geometric verification process.

Depending on the spatial query type, the SP employs different query processing algorithms to retrieve the result. For kNN queries and queries that can be converted into kNN queries (e.g., range queries), query processing can follow any state-of-the-art algorithm that exists in the literature. The resulting VO contains the objects in the result set and their Voronoi neighbor information, which are sufficient for the verification process. For more sophisticated spatial queries, existing query processing algorithms still apply. However, the server needs to return some additional objects so that the client can perform the geometric verification.

5.2 Signature Verification

The client verifies the query result for authentication to proof that the result set is originated from the corresponding DO. For that the client verifies the aggregate signature of VO. When the client receives the VO from the SP, using the public key of the DO the aggregate signature is verify. The verification is simply performing a modular multiplication of the hash digests of all objects included in the VO. Next using the plain text that is derived by decrypting the aggregate signature was compare with the resulted object. If the plaintext is match the objects then the verification is return that the result is correct. Otherwise they conform that the result gets modified by any attackers. If the verification returns fail it stop the further verification otherwise it allow continuing the geometric verification.

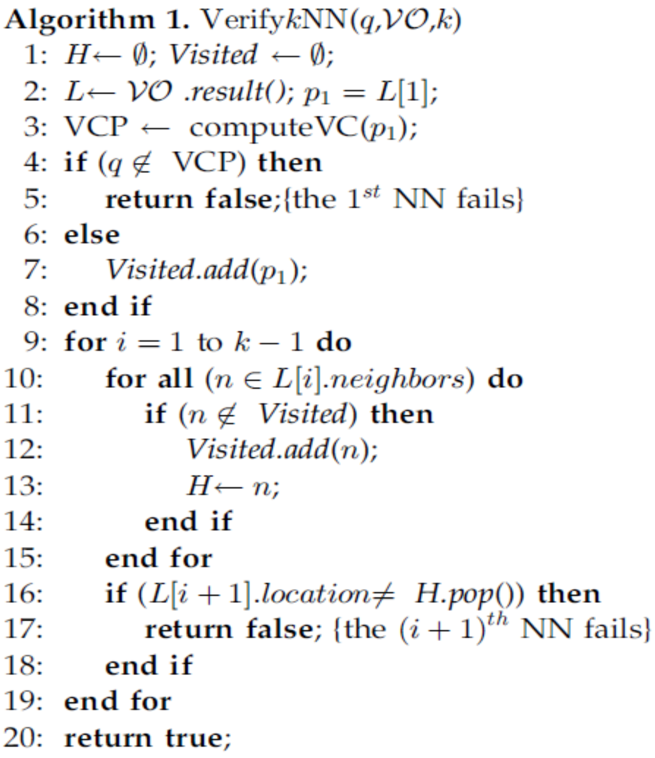
5.3 Geometric Verification

In the geometric verification different type of spatial query has been verified using the addition information about the objects.

5.3.2 KNN and Range Query Verification

Nearest neighbor (NN) queries are the fundamental building blocks in location-based services. In particular, kNN queries allow mobile users to retrieve the k closest POIs from the database, i.e., they may issue queries such as "find the 10 nearest restaurants to my location." For the geometric verification, VN-Auth employs an incremental verification process that is based on Properties 3 and 4. Specifically, according to Property 3, p_i is the first NN of the query point q, if and only if q lies inside the Voronoi cell of p_i. Once this geometric test is verified, Property 4 states that the second NN of q must be one of the Voronoi neighbors of the first NN (p_i). In the general case, the kth NN of a query point q exists in the union of the Voronoi neighbors of the first (k-1) NNs of q.

The subsequent for loop (lines 9-19) iterates through all objects in L (kNNs from the VO) and performs the following operations: 1) if the Voronoi neighbor of the last verified object (L[i]) has not been visited yet, it is inserted into the min-heap H and the Visited set (lines 10-15), and 2) it compares the next object in the result set (L[i+1]) with the top of H (lines 16-18). If they are identical, L[i+1] is verified as the next NN. Otherwise, verification fails and the program returns false. Note that the capacity of the minheap is initially set to (k-1), i.e., it will only hold the first (k-1) objects that are closest to q. Furthermore, the capacity can be decreased by one after each iteration in order to minimize the computational and storage cost of the client.



```
Algorithm 1. Verify$k$NN($q$,$\mathcal{VO}$,$k$)
 1: $H \leftarrow \emptyset$; $Visited \leftarrow \emptyset$;
 2: $L \leftarrow \mathcal{VO}.result()$; $p_1 = L[1]$;
 3: VCP $\leftarrow$ computeVC($p_1$);
 4: if ($q \notin$ VCP) then
 5:     return false;{the $1^{st}$ NN fails}
 6: else
 7:     $Visited.add(p_1)$;
 8: end if
 9: for $i = 1$ to $k - 1$ do
10:     for all ($n \in L[i].neighbors$) do
11:         if ($n \notin Visited$) then
12:             $Visited.add(n)$;
13:             $H \leftarrow n$;
14:         end if
15:     end for
16:     if ($L[i+1].location \neq H.pop()$) then
17:         return false; {the $(i+1)^{th}$ NN fails}
18:     end if
19: end for
20: return true;
```

VI. EXPERIMENTS

6.1 Experimental settings

For VN-Integrity verification the mobile phone is used as client. The verification process is done in the client side and both the Data Owner and the SP are in the same PC. Using the RSA algorithm the DO's sign their data's before send to the SP. The client generate query and send to SP, in the SP query processing take place and send the query result to the client.

VII. CONCLUSION

In this paper, VN-Integrity approach is used to verify the query result. This approach separates the authentication information from the spatial index. Finally , we shows that comparing MR-tree the VN-Integrity produces significantly smaller verification objects, and incurs lower query verification cost and query with low selectivity.

REFERENCES

[1] S. Borzsonyi, D. Kossmann, and K. Stocker. The Skyline Operator. In ICE'01, pages 421-430,2001.
[2] A.Okabe, B. Boots, K. Sugihar and s. chiu. Spatial Tessellations: Concepts and Applications of Voronoi Diagrams. Wiley, second edition, 2000.
[3] H. Pang, J. Zhang and K. Mouratidis. Scalable Verification for Outsourced Dynamic Databases, 2009.
[4] Z. Song and N. Roussopoulos. K-Nearest Neighbor Search for Moving Query point, 2001.
[5] Hacig¨um¨u¸s, H., Iyer, B., Li, C., Mehrotra, S.: Providing Database as a Service. International Conference on Data Engineering (2002).
[6] Narasimha M., Tsudik G. Authentication of Outsourced Databases Using Signature Aggregation and Chaining. DASFAA, 2006.
[7] Sakurai, Y., Yoshikawa, M., Uemura, S., Kojima, H. The A-tree: An Index Structure for High-Dimensional Spaces Using Relative Approximation. Very Large Data Bases Conference (VLDB), 516-526, Cairo, Egypt, September 10-14, 2000.
[8] W. Cheng and K.-L. Tan. Query assurance verication. For outsourced multi-dimensional databases.Journal of Computer Security, 2009.
[9] M. Xie, H. Wang, J. Yin, and X. Meng. Integrity Auditing of Outsourced Data. In VLDB, 2007.
[10] M. L. Yiu, E. Lo, and D. Yung. Authentication of Moving kNN Queries. In ICDE, 2011.