RkNN Query Verification Algorithm

In this section given a query point q and a set of objects P, the reverse kNN (RkNN) query of q is to find all objects in P such that q is one of the k nearest neighbors of p (p2 P). Let RkNN(q) be the result set of q. A verifiable reverse kNN query requests a verification object that contains not only RkNN(q), but also the proof that the RkNN(q) set is correct and complete. Reverse kNN query verification is a challenging problem, especially for verifying that there are no false negatives. This is due to the fact that the cardinality of the result set for a RkNN query is not predetermined by the query.

```
Algorithm: VerifyRkNN(q, VO, k)
1. H ← ∅;Visited ← ∅; R ←VO.result();
2. if (!Verify1NN(p1, q)) then
3. return false;
4. end if
5. VNQ← compute VN(q, p1,p1.neighbors);
6. for all (p VNQ) do
7. P.dl =1;
8. Visited.add(p);
9. H← p;
10. end for
11. while (!H.isEmpty()) do
12. p ← H.pop();
13. i ← getPartition(p);
14. if (L[i].size() == k) then
15. continue;
16. end if
17. if (p∉ VO) then
18. return false;
19. end if
20. L[i].add(p);
21. if (p.dl < k) then
22. for all (n p.neighbors) do
23. if (n ∉ Visited) then
24. n.dl = p.dl+1;
25. H← n;
26. Visited.add(n);
27. end if
28. end for
29. end if
30. end while
31. for all (p L) do
32. if (Verify kNN(p, q, VO)) then
33. if (p ∉ R) then
34. return false;
35. end if
36. end if
37. end for
```

The verification algorithm first checks whether all candidates in P for the RkNN query are returned in the VO. Candidate objects are verified in ascending order of their distance to q. Recall that each object is augmented with its Voronoi neighbors in a clockwise (or counterclockwise) order. To compute the Delaunay distance from q to each p∉ P, we need to verify the first NN p1 of q by checking whether q∉VC(p1). Next, the neighbors of q are computed based on p1 and its neighbors, and the result contains all objects with DL(q, p) = 1 in DG U P {q}). In the example, p1-p4 are identified as the Voronoi neighbors of q and are pushed into a min-heap H where points are sorted according to their distance to q. For now, in each partition a list is used to keep track of the verified candidate objects. Assume that Li be the candidate list for partition Si.

Li contains no more than k objects which are also sorted by their distance to q. After that, we iterate over the top entry p of H (lines 11-30). If partition Si to which p belongs already contains k objects, p is dropped. Otherwise, p is one of the kNNs in partition Si, and we will check whether p is returned in the VO. If p 2 VO, it is inserted into Li (p Si) and all the "not visited" neighbors of p are pushed into H if DL (p, q) < k (lines 21-29). This ensures that every object in H has a DL distance of no more than k. The loop stops when H is empty. If all objects inserted into Li are included in the VO, all candidates for the RkNN query are verified. Finally, for each object p in the candidate set, the VerifykNN algorithm is called to check whether p is a reverse k nearest neighbor of q (lines 31-37).

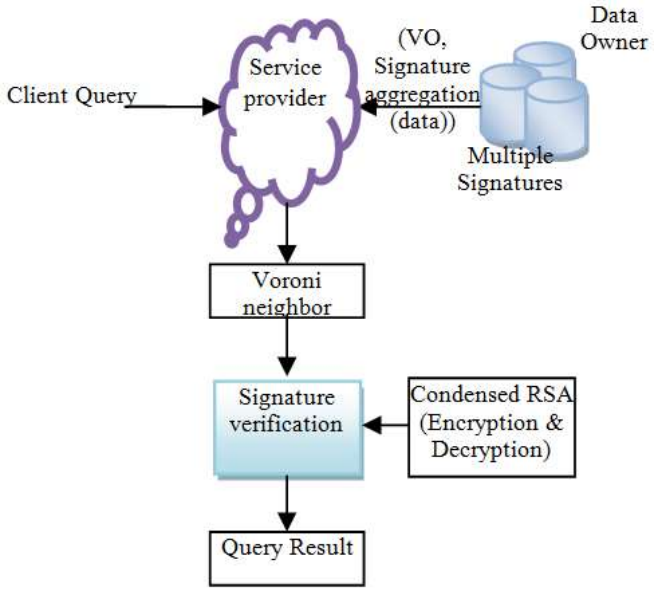

Figure 3: Proposed System Architecture Diagram

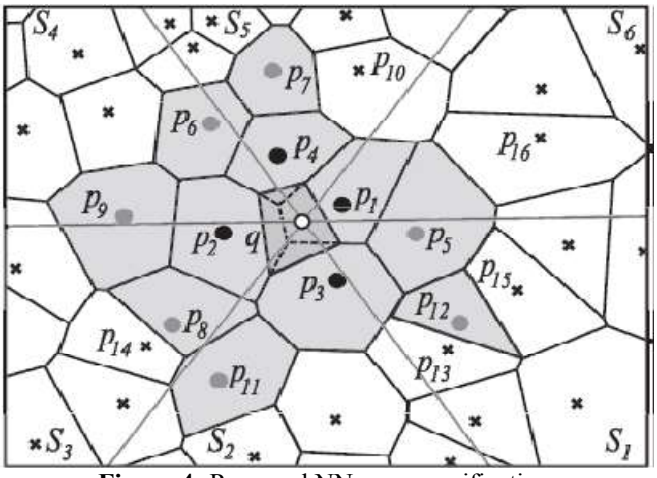

Figure 4: Reverse kNN query verification

4. Experimental Result

The performance of VN-Auth experimentally and compare it against the MR-tree variants. Our implementation is in Java, with the client-side application running on a Google Android mobile device, and the server-side (SP) service hosted on a Windows Server PC with an Intel Core2 Duo 3 GHz CPU and 4 GB memory. The DO also runs on a PC with the same configuration. To implement the cryptographic operations, we used the Java cryptography extension packages.

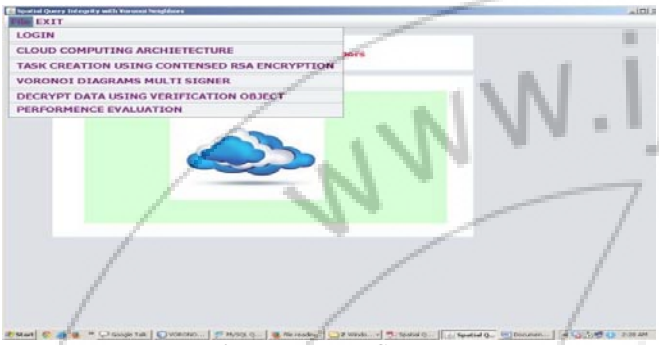

Figure 5: User Screen

The figure 5 shows the users screen which consist of login, cloud computing architecture, Task Creation using RSA, VORONOI diagram multi signer, Decrypt Data Using Verification Object and performance evaluation tabs.

Figure 6: Task Submission Screen

This figure 6 shows the task submission screen after the task creation process where the task is selected, public and private keys are given by user. Then process or condensed RSA encryption is done.

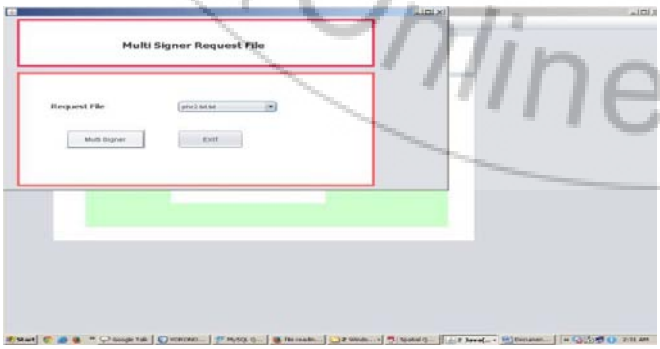

Figure 7: Multi Signer Request File Screen

Figure 7 shows the request of multi signer file given as a txt file as a multi signer. Then the task is encrypted based on it and query request is send to cloud server.

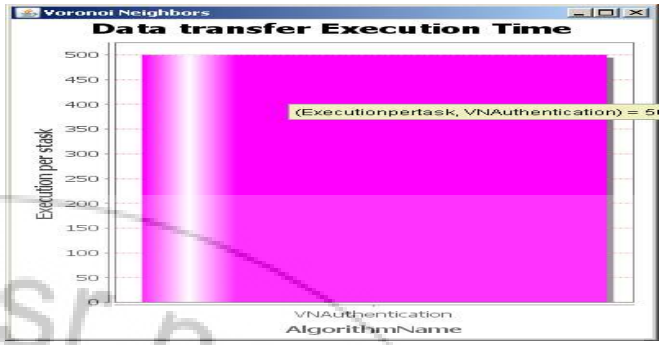

Figure 8: Execution Time taken by Proposed Technique

The figure 8 shows the execution time taken by the proposed technique which is 500 seconds per task.

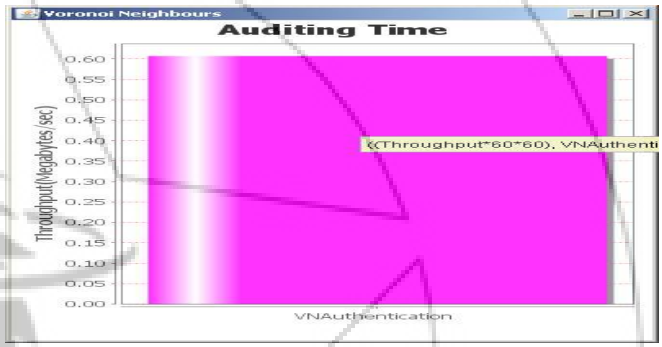

Figure 9: Throughput Time taken by Proposed Technique

Figure 9 shows the throughput of proposed technique takes 0.60 megabytes per second. They are taken based on auditing time.

5. Conclusion

In Outsourced Spatial Database (OSDB) model the clients are mobile users who issue location-based queries (e.g., kNN or range queries), in order to discover points of interest (POIs) in their neighborhood. The MRtree is essentially an R-tree that is augmented with authentication information. kNN queries from mobile clients usually have low selectivity (i.e., the user is interested in very few results) but require fast response time. VN-Auth is a novel approach that successfully meets these requirements in the location-based services model. For location-based services on mobile devices, VN-Auth is clearly a better solution. Query verification can be performed in an incremental fashion, and results can be shown to the end user progressively. In addition, the application may allow the user to examine each batch before receiving the next one. In this way, if the user is satisfied with the current result set and does not wish to retrieve more objects, query processing may be terminated early. The notion of immutability for aggregated signature schemes is used here. Immutability refers to the difficulty of computing new valid aggregated signatures from a set of other aggregated signatures. This is an important feature, particularly for outsourced databases, as lack thereof would enable a frequent querier to eventually amass enough aggregated signatures to answer other (un-posed) queries,

thus becoming a de facto service provider. The proposed authentication schemes have applications such as in healthcare databases, and in authentication of query results of biological and scientific databases. In future, we plan to apply this scheme to some of those domains, as well as in leakage-free assurance of data authenticity in cloud computing.

References

[1] Sion R.: Query execution assurance for outsourced databases. VLDB (2005)
[2] Hacıgümü¸s, H., Iyer, B., Mehrotra, S.: Providing databases as a service. ICDE (2002)
[3] Hacıgümü¸s, H., Iyer, B., Li, C., Mehrotra, S.: Executing SQL over encrypted data in the data-service-provider model. SIGMOD (2002)
[4] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Order-Preserving Encryption for Numeric Data. In SIGMOD, 2004.
[5] E. Damiani, S. D. C. Vimercati, S. Jajodia, S. Paraboschi, and P. Samarati. Balancing Confidentiality and Efficiency in Untrusted Relational DBMSs. In CCS, 2003.
[6] S. Papadimitriou, F. Li, G. Kollios, and P. S. Yu. Time Series Compressibility and Privacy. In VLDB, 2007.
[7] H. Pang, J. Zhang, and K. Mouratidis, "Scalable Verification for Outsourced Dynamic Databases," Proc. VLDB Endowment, vol. 2, no. 1, pp. 802-813, 2009.
[8] S. Papadopoulos, Y. Yang, S. Bakiras, and D. Papadias, "Continuous Spatial Authentication," Proc. 11th Int'l Symp. Advances in Spatial and Temporal Databases (SSTD), pp. 62-79, 2009.
[9] W. Cheng, H. Pang, and K.-L. Tan, "Authenticating Multidimensional Query Results in Data Publishing," Proc. 20th IFIP WG 11.3 Working Conf. Data and Applications Security (DBSec), pp. 60-73, 2006.
[10] W. Cheng and K.-L. Tan, "Authenticating kNN Query Results in Data Publishing," Proc. Fourth VLDB Conf. Secure Data Management, pp. 47-63, 2007.
[11] W. Cheng and K.-L. Tan, "Query Assurance Verification for Outsourced Multi-Dimensional Databases," J. Computer Security, vol. 17, no. 1, pp. 101-126, 2009.
[12] M. Naor. Deniable ring authentication. In Proceedings of Crypto 2002, volume 2442 of LNCS, pages 481{98. Springer-Verlag, 2002.
[13] K. Ohta and T. Okamoto. Multisignature schemes secure against active insider attacks. IEICE Trans. Fundamentals, E82-A(1):21{31, 1999