International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Maintenance of User Query Integrity and Voronoi Neighbors Using Multiple Signers in Signature Aggregation

V. Maruvoor Arasi[1], Carolene[2]

Bharathiar University, Coimbatore, India

Abstract: ***In recent days, the amount of information generated has grown fastly. With that large amount of information in addition to the complexity of the data, require refined management schemes that are ahead of the capabilities of many small businesses or persons. Over the past few years outsourcing spatial data has developed rapidly with the popularity of location-based services and the plentiful usage of smart phones and GPS enabled mechanisms. As a result, the outsourcing database paradigm is more and more all the rage that has received a lot of concentration in the research area. In this model the data owner (DO) entrusts the management and preservation of its database to a third-party service provider (SP) which is accountable for indexing the data and responding client queries from anywhere. In this work, proposes an effective method on the Outsourced Spatial Database (OSDB) model called VN which permits a client to authenticate the accuracy and unity of the result set based on user query. An interactive protocol between the server (Prover) and the querier (Verifier) that gives the concluding with zero-knowledge evidence that the Prover has a valid Condensed-RSA signature in line with the records in the query result set. The server returns to the querier the result set along with a witness. The querier then sends a random challenge to which the server responds with a valid response to the client. The reply among the witness induces the querier of server's knowledge of the Condensed-RSA signature, devoid of disclosing any information about the Condensed-RSA signature itself. Additionally, present the multisignature scheme that is provably secure devoid of random oracles where a single short object the multisignature can substituten signatures by n signers that all on the same message.***

Keywords: Query Authentication, Spatial Queries, Outsourced Databases, Condensed-RSA Signature, Multi Signature Scheme, Voronoi Diagrams

1. Introduction

Database outsourcing is a new model that has been suggested in recent times and has received a lot of attention in the research field. Mainly there are three entities in the Outsourced Database (ODB) model such as the data owner, the client and the database service provider; actually, there is a single or a few data owners, a few servers, and many clients will be available. With the associated index and authentication structures the data owner generates the database and uploads to the servers in ODB. Periodically or occasionally the data owner may revise the database, as well as that the data organization and recovery happens only at the servers. The fundamental design is that data owners (DO) assign their database protection and functionalities to a third-party service provider (SP), as well as the SP is accountable for indexing the data and responding client queries. While the third party can be untrusted or can be cooperated, security concerns must be tackled previous to this designation.

Clients might submit queries about the owner's data to the servers and get back outcomes during the network. On the whole, in distributed environment it is economical to preserve usual servers than to preserve truly secure server. The owner has to provide the clients the capability to validate the answers they accept devoid of having to hope the servers on the network to safeguard against malicious or compromised servers. Given that respect the query authentication has two significant quantities: completeness and correctness. Exactness means that the client must be able to authenticate that the revisited records do exist in the owner's record and have not been adapted. Completeness means that no responds have been skipped from the consequence which is revisited back to the client. Suppose that the clients are mobile users who subject location-based queries (e.g., kNN or range queries), so as to determine summits of interest (POIs) in their area. As the SP is not the real owner of the data, query integrity assurance is a demanding crisis that has to be tackled.

Especially, the SP has to establish to the client that (i) the data is created from the DO and, (ii) the result set is right and absolute. The general construction that is frequently utilized is based on digital signatures and uses a public-key cryptosystem, for example RSA. At first, DO obtain a private and a public key, throughout a trusted key distribution center in cryptosystem. The public key is available by all the clients and private key is reserved secret at the DO. The DO digitally signs the data through generating signatures using private key. After that, it sends the signatures and the data to the SP that builds the necessary data structures for well-organized query processing model. It produces a verification object (VO) when the SP obtains a query request from a client that encloses the result set along with the essential authentication data. Finally, the SP sends the VO to the client that can confirm the consequences by means of the public key of the owner.

The structure of the MR-tree as well as the verification procedure, endure from various drawbacks. Primary, the verification information (hash digests) embedded in the MR-tree reduces the node disperse, directing to more I/O accesses throughout processing of query. Before improves following, the entire processes on the path from manipulated leaf node to the root have to be reevaluated. Query performance is mortified when updates are frequent.

In conclusion, especially for queries that return only a few objects the overhead of the VO can be significant. This is

because of the fact that SP has to revisit all objects that lie within the leaf nodes that are stayed throughout processing of query. MR*-tree, an expansion of the MR-tree, alleviates this disadvantage, through arranging the entries of every node and makes hierarchical relationships of assimilates there. It augments the verification cost at the client, does not reduce the VO overhead completely, even as at the similar instance. In this work suggested Merkle Hash Tree (MHT)-based advances with VN-auth to decrease the time difficulty when query selectivity is elevated.

The Certification authority created and contracted the Merkle Hash Tree (MHT) and then dispersed to untrusted directory services. Units needing to authenticate the cogency of a certificate can demand such a directory service and have assurance in the exactness of a reply by using the returned data to confirm the certificate authority's sign. Moreover it lessens the difficulty of the structure. In the extreme case where the client regains the complete database where the MHT approach would only return the root signature. Whereas VN-Auth would need to return 6.n neighbors, where n is the database cardinality. An communicating protocol among the server (Prover) and the querier (Verifier) that delivers the latter with a zero-knowledge proof that the Prover has a effective Condensed-RSA signature conforming to the records in the query result set.

With a witness the server returns to the querier the result set along. With a valid response the querier then sends a random challenge to which the server replies. The response together with the witness convinces the querier of server's knowledge of the Condensed-RSA signature, without revealing any knowledge about the Condensed-RSA signature itself. The remainder of the paper is organized as follows. Section 2 talks about the related work. Section 3 describes signature aggregation techniques. Section 4 explains the data revolution process, and Section 5 describes signature verification and introduces MHT for authentication. Section 6 concludes the paper with directions for future work.

2. Related Work

Several papers study privacy preservation of outsourced data, under a model similar to [1], i.e., the only client is the DO. Hacıgumus et al. [2, 3] assume that the DO transmits encrypted data to the server, along with a set of crypto indices individual per query attribute. In particular, for every such attribute, the DO partitions all its values into buckets, and stores in the corresponding crypto index the bucket ids for each record. To process a range query Q, the DO translates Q into Q′by replacing attribute values with bucket ids; the server answers Q′ using the crypto indices and returns the encrypted tuples. Note that the results of Q′ are a superset of that of Q. Decryption and filtering of false hits are performed at the client's site.

In a proposal by Damiani et al. [4], the data owner builds a B+ tree on one dimensional data and then encrypts each node using conventional cryptographic techniques (e.g., AES). While such encryption does not preserve order, the decryption key is required in order to process queries. A trusted front end at the SP is assumed. In practice, a tamper resistant device can be used for this purpose. Our CRT technique uses a similar approach in conjunction with an R tree for two dimensional data. In our case, though, the SP is likely to serve a large number of data owners, each with a distinct encryption key. In addition the SP may tender the service for free, interpretation it monetarily infeasible to employ one tamper resistant device per data owner. For this reason, the SP and the user must employ a multi stage protocol during query processing; therefore, we must minimize the communication cost.

Papadimitriou et al. [5] apply perturbation techniques on time series data before they are published. Correlated noise is added into the time series such that (i) the utility of the perturbed data is bounded by a specified threshold, and (ii) the error between the adversaries's reconstructed data and the original data is maximized. They consider two attack models: filtering and true value leakage. These attacks are analogous to the noise removal attack and the general attack. The above technique is inapplicable to our problem because there is no key for accurately reconstructing the original data from the perturbed data.

Efficient verification in the presence of frequent updates has been studied in the context of relational data. The Partially Materialized Digest scheme (PMD) [6] verifies 1D query and applies to both static and dynamic databases. Similar to our proposed approach, PMD employs separate indexes for the data and their associated verification information in order to avoid unnecessary costs when processing queries that do not request verification. Moreover, the authors designed two verification methods for spatial queries namely the Merkle R tree and the Partially Materialized KD tree (PMKD). The first is an extension of the MBtree method [4] to R tree indexes, and the second an adaptation of the PMD methodology for spatial data.

Furthermore, Pang et al. [7] introduced a protocol, based on signature aggregation, which verifies the authenticity, completeness, and freshness of the query consequence. A significant property of the protocol is that it allows new data to be disseminated immediately, while ensuring that outdated values (beyond a preset age) can be detected. In addition, the authors also implemented an efficient verification technique for ad hoc equijoins. Papadopoulos et al. [8] designed a solution for the authentication of continuous spatial queries, i.e., queries that are constantly evaluated on a highly dynamic database (consisting of moving objects). The proposed mechanism achieves both correctness and temporal completeness and aims at reducing the transmission overhead between the service provider and the clients.

Based on [9], Cheng and Tan designed a mechanism for authenticating kNN queries on multidimensional databases, ensuring that the result set is complete, authentic, and minimal [10], [11]. Nevertheless, both solutions incur significant authentication overhead, and the required verification information consumes considerable client server communication bandwidth. Yang et al introduced the MR and MR* trees, which are space efficient authenticated data structures sup porting fast query processing and verification. The MR tree augments the standard R tree, by computing hash digests on the concatenation of the binary representation of all the entries in a tree node. To verify the correctness and

completeness of range query results, the generated VO includes (i) all the visited objects, and (ii) the MBRs and digests of all the pruned nodes. The MR* tree improves MR tree by ordering the entries of each node and constructing hierarchical relationships of the digests therein. Entries are sorted according to an in order traversal of a KD tree. As a result, when a query intersects an MBR, not all entries are required for query verification, and some of them can be shortened. The thought is similar to building a small Merkle tree on each node of the MR tree. The MR* tree reduces significantly the VO size, but incurs some CPU overhead due to the embedded information.

3. Proposed Methodology

1.1 Voronoi Diagrams for Multi Signature

We contend that minimizing searching response time is important to mobile occurrence services. Consequently, the buddy list searching algorithm of Presence Cloud coupled with the two-hop overlay and one-hop caching strategy ensures that Presence Cloud can typically provide swift responses for a large number of mobile users. Initially, , we can therefore use one-hop search exactly for queries and thus reduce the network traffic without significant impact on the search results through organizing PS nodes in a server-to-server overlay network. Second, by capitalizing the one-hop caching that maintains the user lists of its neighbors, by increasing the chances of finding buddies we improve response time. Clearly, this mechanism both reduces the network traffic and response time. The population of mobile users can be retrieved by a broadcasting operation in any PS node depending on the mechanism in the mobile occurrence service. Furthermore, the broadcasting message can be piggybacked in a buddy search message for saving the cost. A Voronoi diagram is a way of dividing space into a number of regions. A set of points that is called generators, seeds and sites that is specified beforehand and for each seed there will be a corresponding region consisting of all points closer to that seed more than any, regions are called Voronoi cells which is dual to the Delaunay triangulation.

Given a set of distinct objects $p = (p_1, p_2, \ldots, p_n)$ in R_m, the Voronoi diagram of P, denoted as WD(P), partitions the space of Rm into n disjoint regions, such that each object pi in P belongs to only one region and every point in that region is closer to pi than to any other object of P in the Euclidean space. The region around pi is called the Voronoi cell of pi, denoted as $VC(p_i)$, and pi is the generator of the Voronoi cell. Therefore, the Voronoi diagram of P is the union of all Voronoi cells. Service providers process queries on the outsourced database (the database stored in the cloud) on behalf of the data owner. For verifiable queries, returning the query result to the clients is no longer sufficient. Instead, the SP is required to return a verification object that contains 1) a condensed signature AASS that verifies the authenticity of all objects in the VO and 2) the result set of the query with some additional objects that are necessary for the geometric verification process. The resulting VO contains the objects in the result set and their Voronoi neighbor information that are enough for the verification procedure for more sophisticated spatial queries. The server needs to return some additional objects so that the client can perform the geometric verification.

1.2 Multisignatures Scheme

In a multisignature scheme, a single multisignature the same size as one ordinary signature stands for l signatures on an M message. Multisignatures were initiated by Itakura and Nakamura that has been the subject of much research. The first multisignatures in which signatures could be combined into a multisignature without interaction was proposed by Boldyreva, based on BLS signatures. Below, we present another non-interactive multisignature scheme, based on the Waters signature scheme without random oracles that is truly secure.

A multisignature scheme comprises five algorithms; three of these are Kg, Sig, and Vf that analogous to those in ordinary signature schemes, randomized key-generation algorithm Kg outputs a public {private keypair (pk, sk). The randomized signing algorithm Sig takes a private key sk and a message $M \in \{0,1\}^*$ and outputs a signature σ. The verfication algorithm Vf takes a public key pk, signature σ and a message M. The outputs a bit: 1 if the signature is legal, 0 or else. The two remaining algorithms provide the multisignature functionality. The first, Comb, all on a common message M combines l ordinary signatures where each under a different key, into a single multisignature that stands for all the input signatures. More formally, Comb is a randomized algorithm that takes the l public key{signature pairs $\{p_k^i, \sigma_i\}_{i=1}^l$ along with the message $M \in \{0,1\}^*$ and outputs a multisignature σ or, if combining the signatures failed, ⊥. We stress that the combination algorithm requires the public keys of all the users, not just the signatures themselves. The second algorithm, MVf, performs multisignature verification. It takes the l public keys $\{p_k^i\}_{i=1}^l$; the common message M; and the multisignature σ that purportedly stands for signatures on M under each of the keys where if the multisignature is valid then outputs a bit: 1, or else 0.

We include the constraint that neither the combination algorithm nor the multisignature verification algorithm allows a single signer's key to appear more than once in the key list $\{p_k^i\}_{i=1}^l$;A multisignature scheme, instantiated using these algorithms, is correct if all properly-generated signatures and multisignatures verify. More formally, for all signer keypairs (pk, sk) and (pki, ski) output by Kg, all messages M, and all $l \geq 1$, the following hold with probability 1:

$$\mathrm{Vf}(\mathrm{pk}, \mathrm{M}, \mathrm{Sig}(\mathrm{sk}, \mathrm{M}) = 1$$

$$\mathrm{MVf}\left(\{p_k^i\}_{i=1}^l, \mathrm{M}, \mathrm{Comb}\left(\{\mathrm{pki}, \mathrm{Sig}(\mathrm{Ski}, \mathrm{M}\}_{i=1}^l, \mathrm{M}\right)\right) = 1$$

Immutable Condensed RSA (IC RSA)

To make condensed RSA signatures immutable, we use the technique that can be broadly classified as a Zero knowledge proof of knowledge of signatures. Instead of revealing the actual aggregated Signature for a posed query, the server reveals only the proof of knowledge of that signature. We

present two variants: one that requires interaction, based on the well known Guillou Quisquater scheme, and the other that is non interactive, based on so called "signatures of knowledge".

Interactive Variant

This technique uses the well known Guillou Quisquater (GQ) identification scheme [12] which is among the most efficient follow ons to the original Fiat Shamir zero knowledge identification Scheme [2]. The version we present is an interactive protocol between the server (Prover) and the querier (Verifier) that provides the latter with a zero knowledge proof that the Prover has a valid Condensed RSA signature corresponding to the records in the query result set.

Basically, the server returns to the querier the result set along with a witness. A random challenge will send by the querier then to which the server replies with a valid response. The response together with the witness convinces the querier of server's knowledge of the Condensed RSA signature, without revealing any knowledge about the Condensed RSA signature itself. The overview of the protocol is shown in Figure 1.

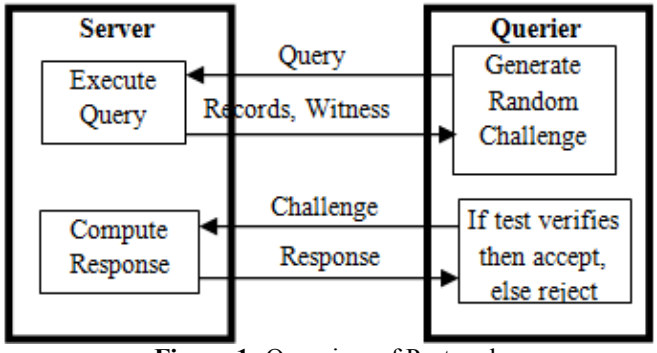

Figure 1: Overview of Protocol

The terms Prover (P) and Verifier (V) is used instead of Server and Querier, respectively, since the protocol is not specific to the ODB setting. Let $X = \sigma_{1,t} = \prod_{i=1}^{t} \sigma_i (\text{mod } n)$ the condensed RSA signature computed as shown above. Recall that (e,n) * is the public key of the original data owner which all concerned parties are assumed to possess. Let $M \equiv \prod_{i=1}^{t} h(m_i) (\text{mod } n)$ and $X^e = (\sigma_1, t)e \equiv M (\text{mod } n)$.

- In first step, the querier poses a query.
- In next step the server (prover) replies with the result set for that query as well as a commitment Y. Note that $Y = r^e$ here is a randomly chosen element, the RSA modulus of the data owner who generated the individual RSA signatures corresponding to the records in the result set 7 and e the corresponding public exponent.
- In this step the verifier (querier) sends back a challenge v that is chosen randomly from $\{0,1\}^{l(k)}$ where $l(k)$ is the bit length of the public exponent e.
- In this Step, server, upon receiving the challenge v, computes the response $z = r\, X^v (mod\ n)$ where X is the Condensed RSA signature of the result set.
- In this step, the verifier accepts the proof if $Z^e \equiv Y\, M^v (\ mod\ n)$ where M is the product of all messages in the result set.

Query Integrity Assurance

In addition to data privacy, a challenging security concern in the database outsourcing is Query integrity. Query integrity examines the guarantee of the hosting environment to the client. It wants to be assured when a client receives a query result from the service provider that the result is both correct and complete. Correct denotes query must be evaluated honestly with the outsourced database to retrieve the result and complete means result includes all the records Satisfying the query. Devanbu et al. proposed to employ the Merkle hash tree authenticate data records. The technique computes a signature based on the Merkle hash tree structure and distributes to clients for correctness. Mykletun et al. studied and compared several signature methods which can be applied in data authentication.

1.3 Signature Verification

In signature verification two steps are taken in spatial query verification process i.e. first, the aggregate signature of the verification object (VO) is observed by the client to ensure that all returned object by the SP Originated at the DO. Second, all objects in the result set are evaluated to ensure the geometric properties are satisfied and no Legitimate objects are eliminated. When a client receives the VO from the SP, it verifies the aggregate signature using the public key of the DO. particularly, a modular multiplication of the hash digests of all object performed by the client simply included in the VO, as well as verifies that result and matches the plaintext which is derived by decrypting the aggregate Signature with the DO's public key. The client considers the result as corrupted and verification process terminates when the VO fails the signature verification process. Otherwise, it continues with the multi signature Verification.

Figure 2: Signature Architecture

In figure 2, 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. However, since the SP is not the real owner of the data, query integrity assurance is an important (and challenging) problem that has to be tackled. Especially, the SP has to confirm to the client that (i) the data is originated from the DO and, (ii) consequence set is correct and complete.

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).

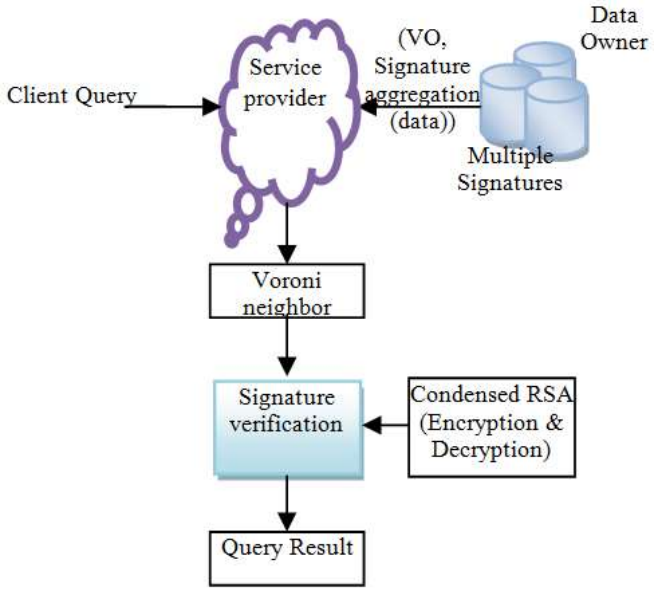

Figure 3: Proposed System Architecture Diagram

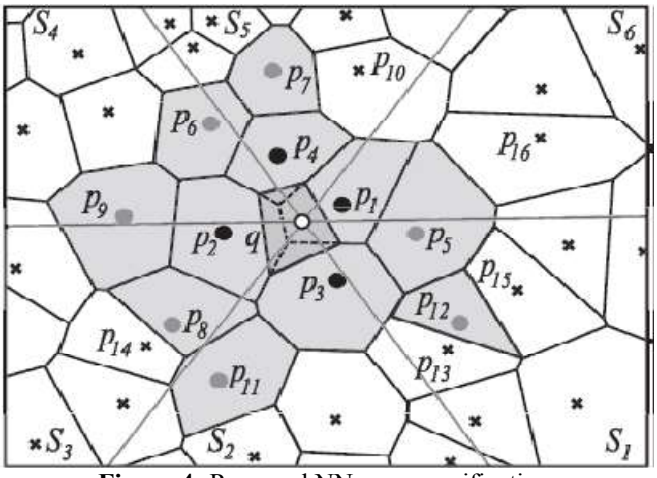

Figure 4: Reverse kNN query verification

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.

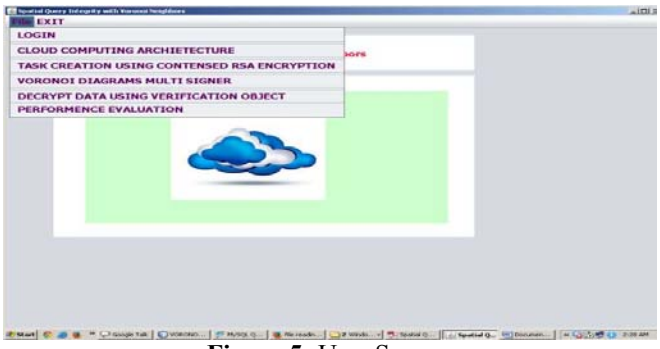

Figure 5: User Screen

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.

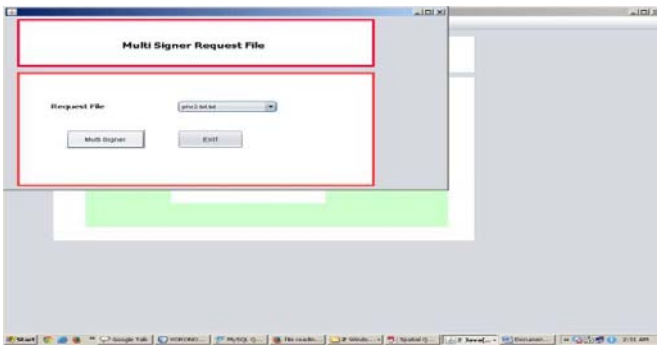

Figure 7: Multi Signer Request File Screen

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.

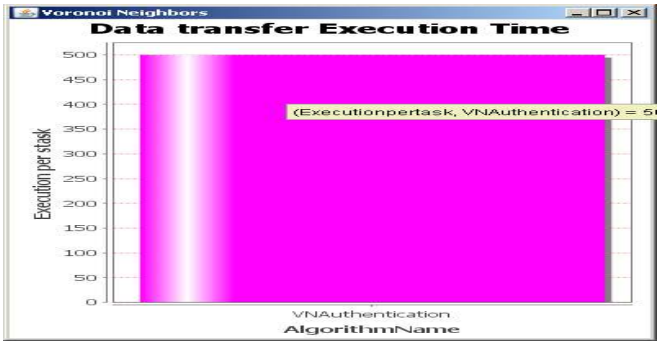

Figure 8: Execution Time taken by Proposed Technique

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

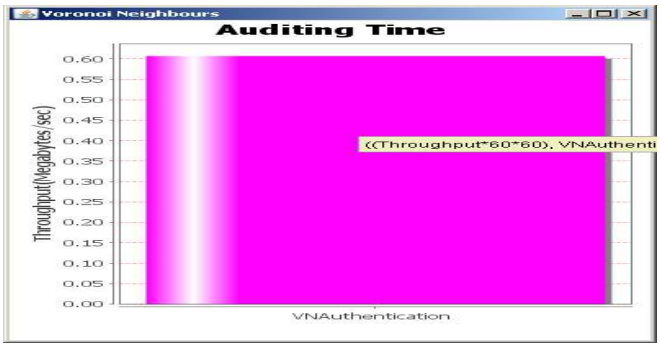

Figure 9: Throughput Time taken by Proposed Technique

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