Location Based Search with Merkle Skyline R Tree Method For Moving Object

P.Senthilraja[1], R.Kousalya[2]
[1, 2] Department of Computer Science and Engineering
Assistant Professor,R.V.S. Educational trust's group of institutions, Dindigul, India.
M.Phil scholar,Cardamom Planter's Association College, Bodinayakanur ,India

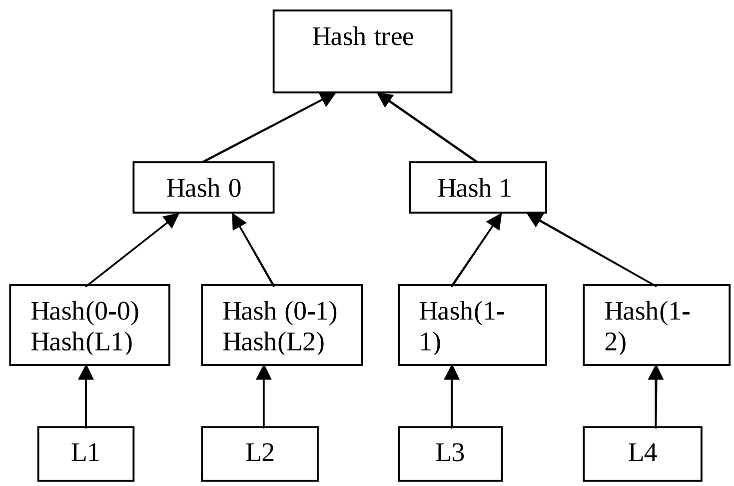
Abstract*-Location-based services are used to the client carrying location aware mobile devices are able to query Location based services for surrounding points of interest anywhere and at any time. The many types of location-based queries, one important class is location-based skyline queries(LASQ) For the authentication of continuous LASQs, we develop a perfecting-based approach that enables clients to compute new LASQ results locally during movement, without frequently contacting the server for query re-evaluation. Experimental results demonstrate the efficiency of our proposed methods and algorithms under various system settings. We also proposed the work to road network environments. The query distance is defined by network distance in a road network, the skyline scope defined in this method no longer works, which calls for new authentication methods.*

Keywords*-*Merkle Skyline, LASQ queries.

I. INTRODUCTION

LASQ scale up the Location-based services along with their ever-growing popularity a rising trend is to outsource data management and service provisioning to Cloud service provider such as Amazon EC2 and Google App Engine. The data owner obtains through a certificate authority a pair of private and public keys of digital signatures. Before delegating a spatial dataset to the Cloud service provider the data owner builds an authenticated data structure of the dataset. To support efficient query processing, the authenticated data structure is often a tree-like index structure, where the root is signed by the data owner using his/her private key. The Cloud service provider keeps the spatial dataset, as well as the authenticated data structure and its root signature. Upon receiving a query from the client, the Cloud service provider returns the query results, the root signature, and a verification object which is constructed based on the authenticated data structure. The client can authenticate the correctness of the query results using the returned verification object the root signature, and the data owner's public key. To extend this study to the general problem of authenticating location-based skyline queries in arbitrary subspaces of attributes. Because a basic solution that returns all results in the full space is inefficient, it proposes a new authentication method based on the notion of signed sub-space skyline scope of partial S4 tree. We devise a data structure, called Partial-S4-tree, which precomputes, signs, and stores the skyline scopes of some subspaces. For continuous LASQs, the concept of clear area is introduced to enable a moving client to re-evaluate new results locally. Moreover, we propose an approach to prolong the client's residence time inside a clear area.

Merkle Tree



In cryptography and computer science, a hash tree or Merkle tree is a tree in which every non-leaf node is labelled with the hash of the labels or values (in case of leaves) of its child nodes. Hash trees are useful because they allow efficient and secure verification of the contents of large data structures. Hash trees are a generalization of hash lists and chains. Demonstrating that a leaf node is a part of the given hash tree requires processing an amount of data proportional to the logarithm of the number of nodes of the tree this contrasts with hash lists, where the amount is proportional to the number of nodes. Hash trees can be used to verify any kind of data stored, handled and transferred in and between computers. Currently the main use of hash trees is to make sure that data blocks received from other peers in a peer-to-peer network are received undamaged and unaltered, and even to check that the other peers do not lie and send fake blocks.

Suggestions have been made to use hash trees in trusted computing systems.

Location Based Query Authentication

We propose a basic LASQ authentication method. We start with the authentication problem in a fixed subspace, and then extend it to arbitrary subspaces.

LASQ Authentication

Design of Authenticated Index Structure. To expedite query processing, we index all the objects' subspace skyline scopes by an R*-tree , where the subspace skyline scopes are stored in the leaf nodes as data entries. Additionally, to support query authentication, we follow similar ideas of MB-tree and MR-tree to maintain a series of digests for all index nodes in the tree structure.

Server Query Processing and VO Construction. With the help of MSR-tree, an LASQ is reduced to a point-location Apply Divide and Conquer method to Block Nested Loop.

Block Nested Loop(BNL)
Divide and Conquer(D&C)

Block Nested Loop

BNF is faster to computing skyline query.

When tuples P is read from the input, P is compared to all tuples

Three conditions,
P is dominated by a tuples – P is eliminated.
P is dominates other tuples – P is eliminated.
P is incomparable with all tuples – P is read.

Divide and conquer

begin
if base case then
solve problem
else
partition problem into sub problem L and R
solve problem L using D&C
solve problem R using D&C
combine solution to problem L and R
end if
end

Data Owner

Merkle Tree the digests of index nodes are recursively computed from the leaf nodes to the root. After that, the root digest is signed by the data owner's private key and stored on the outsourced database server.

Verification Object

The verification object (VO) is used to verify the query detail with contact of the data owner. The data owner provides has to store the query. The redundant objects can be easily identified and safely removed from the VO, thereby minimizing its size and saving the server processing time.

II. METHODOLOGY

The research for to find the location automatically moving object with help of location-based arbitrary-subspace skyline queries Less overhead in querying process. Client location can be identified. For the authentication of continuous LASQs, we develop a pre fetching-based approach that enables clients to compute new LASQ results locally during movement, without frequently contacting the server for query re-evaluation. Experimental results demonstrate the efficiency of our proposed methods and algorithms under various system settings. We also proposed the work to road network environments. The query distance is defined by network distance in a road network, the skyline scope defined in this method no longer works, which calls for new authentication methods.

Authenticated query processing

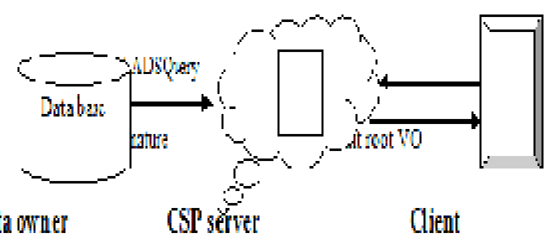


Figure 3.4. Authenticated Query Processing

Client Detail

The client has sign the login page and register in the application, then the application find client location automatically. Further we can search the location for our needs. The client query are stored in the data owner private key and the out source of the server.

Service provider

The service provider is to give detail about the client query. The query are stored in the data owner. It sent the detail about the query to the data owner.

Skyline Query Processing

A number of algorithms have been developed since then. These algorithms can be divided into two categories. The first category is non-index-based algorithms. The representatives are Block-Nested-Loop (BNL) and Divide-and-Conquer (D&C). The CSP may return incorrect results unintentionally because of bugs in the implementation of query processing algorithms. The CSP (or the adversary who compromised it) may intentionally tamper with the query results. The other category of skyline algorithms is index-based. A high-dimensional dataset is converted into a one-dimensional dataset and a B + -tree is built to accelerate query processing.

Skyline Query processing using two algorithms

Block Nested Loop(BNL)
Divide and Conquer(D&C)

BNF is faster to computing skyline query

Apply Divide and Conquer method to Block Nested Loop

Block Nested Loop

When tuple P is read from the input, P is compared to all tuples

Three conditions,

P is dominated by a tuple – P is eliminated

P is dominates other tuples – P is eliminated

P is incomparable with all tuples – P is read.

```
Find-max-crossing-subarray(A,low,mid,high)
//Find a maximum subarray of the form A[i..mid].
Left-sum=-∞
Sum = 0
For i = mid downto low
        Sum = sum +A[i]
        if sum > left-sum
        left-sum = sum
        max-left = i
//Find a maximum subarray of the form A[mid+1…j]
right=sum=-∞
sum=0
for j = mid + 1 to high
        sum = sum + A[j]
        if sum > right-sum
        right-sum = sum
        max-right = j
//Return the indices and the sum of the two subarray.
return (max-left,max-right,left-sum+right-sum)
```

Divide and Conquer algorithms

Divide and Conquer algorithms split a problem into separate sub problems, solve the sub problems and combinethe results for a solution to the original problems. Divide anConquer algorithms is a top-down approach algorithm.

Signature Generation

We identify the problem of authenticating LASQs in outsourced databases. To the best of our knowledge, this study is the first attempt to investigate this problem. We developed new schemes for range and top-k query authentication that preserves the location privacy of queried objects. The data owner obtains, through a certificate authority (e.g., VeriSign), a pair of private and public keys of digital signatures. Before delegating a spatial dataset to the CSP, the data owner builds an authenticated data structure (ADS) of the dataset. To support efficient query processing, the ADS is often a tree-like index structure, where the root is signed by the data owner using his/her private key. The CSP keeps the spatial dataset, as well as the ADS and its root signature. Upon receiving a query from the client, the CSP returns the query results, the root signature, and a verification object (VO), which is constructed based on the ADS. The client can authenticate the correctness of the query results using the returned VO, the root signature, and the data owner's public key.

Tree Generation

Merkle Skyline R-tree method and a Partial-S4-tree method, aiming to reduce the server processing time and minimize the VO size. The root is b's full-space skyline scope, which serves as a data entry in the original MSR-tree. Each edge in the Partial-S4-tree is labelled with the attribute(s) by which the subspaces of the parent and the child node differ. The final VO is composed of: 1) the VO-tree constructed based on the full-space MSR-tree (excluding the redundant objects and their full-space skyline scopes ($So,_{\wedge}$'s)); 2) the subspace skyline scopes ($So,_{\wedge}$''s) of all redundant objects and their aggregate signature.

LASQ Authentication

We propose a basic LASQ authentication method. We start with the authentication problem in a fixed subspace, and then extend it to arbitrary subspaces. Where the subspace skyline scopes are stored in the leaf nodes as data entries. Additionally, to support query authentication. The digest of each index node can be computed recursively in a bottom-up fashion. Finally, the digest of the root node is computed and signed by the data owner with his/her private key to generate the root signature, Sig (Hroot). Hereafter, this authenticated index structure is called Merkle Skyline R-tree. Server Query Processing and VO Construction. With the help of MSR-tree, an LASQ is reduced to a point-location query on the indexed subspace skyline scopes. Specifically, starting from the root and going all the way down to the leaf nodes, the server checks whether any child of a node covers the query point. If it does, the node is unfolded and inserted into the VO for further checking; otherwise, the node is pruned and only its MBR and digest are inserted into the final VO. When visiting a leaf entry associated with an object t_o, if the corresponding S_o does not cover the query point, both S_o and H_o should be inserted into the VO; otherwise, *o* is an LASQ result and only So is inserted into the VO (H_o can be computed locally by the client based on the received result). It is noteworthy that as the nodes in the VO also form a tree structure, we call it a VO-tree.

III. CONCLUSION AND FUTURE ENHANCEMENT

The proposed system focuses the problem of authenticating location-based Arbitrary subspaces skyline Queries (LASQs). MSR-tree authentication method can be proposed by extending our work on skyline query authentication. To enable authentication for large-scale datasets and subspaces, further implementation of Partial-S4-tree method, in which most of the redundant objects can be easily identified and filtered out from the VO. For authenticating continuous LASQs, the system has pre fetching-based solution to avoid frequent query issuances and VO transmissions. Extensive experimental results demonstrate that the performance of merkle tree methods. Further it can work in large data set and also query processing time is low. The merkle tree algorithm perform well in the location search methods.

FUTURE ENHANCEMENT

In future we can work this technique to the traffic environment. The query distance is defined by network distance in a road Network. The skyline scope defined in this work no longer workswhich calls for new authentication methods.

REFERENCES

[1] Yan X. and Han J. (2005), ‘Substructure similarity search in graph databases’, pp.766-777.

[2] Zhu Y. Qin L. Yu J.X. and Cheng H. (2012), ‘Finding top-k similar graphs in graph databases’, pp.456-467.

[3] He H. and Singh A. K. (2006), ‘Closure-tree: An index structure for graph queries’, pp38-38.

[4] Yuan yuan T. Carlos C. S. States D. J. and Patel J. M. (2007), ‘SAGA: A subgraph matching tool for biological graphs’, Vol.42, No.2, pp.232-239.

[5] Pang H. Jain A. and K Tan K.L. (2005), ‘Verifying completeness of relational query results in data publishing’, pp.407-418.

[6] Kundu A. and Bertino E. (2010), ‘How to authenticate graphs without leaking’, pp.609-620.

[7] Martel C. Kwong A. and Stubblebine S.G. (2004), ‘A general model for authenticated data structures’, Vol.39, No.1, pp.21-41.

[8] Khan A. Li N. Yan X. Guan Z. and Tao S. (2011), ‘Neighbourhood based fast graph search in large networks’, pp.901-912.

[9] Beckmann N. Kriegel H.P. Schneider R. and Seeger B. (1990), ‘The R*-tree: An efficient and robust access method for points and rectangles’, pp. 322–331.

[10] Huang Z. Lu H. Ooi B. C. and Tong K. H. (2006), ‘Continuous skyline queries for moving objects’, Vol. 18, No. 12, pp. 1645–165.