Asian Journal of Convergence in Technology Volume II, Issue I
Issn No.:2350-1146, I.F-2.71

Secure kNN Query Processing using VD-kNN In an Untrusted Cloud Enviroments

Miss. Monika D. Rokade.
M.E IInd Year
Department of Computer
Sharadchandra Pawar College of Engineering,
Otur, Pune.
monikarokade4@gmail.com

Prof. Sandip A. Kahate.
Assistant Professor
Department of Computer
Sharadchandra Pawar College of Engineering,
Otur, Pune.
sandip.kahate@gmail.com

Abstract*— At the moment, data is stored with a third party in cloud environments and query processing is also done by third party to reduce the amount to maintain the system. Mobile devices with geo-positioning capabilities register users to access information that is applicable to their present and current location. Users are interested in querying about points of interest (POI) in their physical concurrence. e.g. restaurants, cafes, gas station, etc. Objects specialized in various areas of interest (e.g., entertainment, and travel) gather large amounts of geo-tagged data that appeal to register users. Such data may be observant due to their substance. Furthermore, keeping such information latest form and applicable to the users is not an easy task , so the holder of such datasets will make the data available only to paying customers. Users send their current location as the query parameter and wish to receive as result the nearest POIs, i.e., nearest-neighbors (NNs).

Keywords *—Query services; kNN query; structural databases; mutable order preserving encoding.*

I. INTRODUCTION

We propose several schemes. First, secure kNN query processing and secure proximity detection, which is based on Mutable Order Preserving Encryption (MOPE) and Secure Point Evaluation Method (SPEM). Second, for authenticated top-k aggregation, we suggest new method of using Three Phase Uniform Threshold Algorithm, Merkle Hash Tree, and Condensed-RSA. Third, for detecting malicious nodes, we propose new algorithms based on Additively Homomorphic Encryption and Multipath Transmission. Experimental evaluation and security analyses demonstrate that robust mechanisms can be deployed with a minimal amount of computational and communicational expense. Emergence of mobile devices with fast Internet connectivity and geo-positioning capabilities has led to a revolution in customized *location-based services (LBS)*, where users are enabled to access information about *points of interest (POI)* that are relevant to their interests and are also close to their geographical coordinates. Probably the most important type of queries that involve location attributes is represented by *nearest-neighbor (NN)* queries, where a user wants to retrieve the *k* POIs (e.g., restaurants, museums, gas stations) that are nearest to the user's current location (*k*NN) [1].

II. RELATED WORK

A vast amount of research focused on performing such queries efficiently, typically using some sort of spatial indexing to reduce the computational overhead [1]. The issue of privacy for users' locations has also gained significant attention in the past. Note that in order for the NNs to be determined, users need to send their coordinates to the LBS. However, users may be reluctant to disclose their coordinates if the LBS may collect user location traces and use them for other purposes such as profiling, unsolicited advertisements, etc. To address the user privacy needs several protocols have been proposed that withhold, either partially or completely the users' location information from the LBS. For instance, the work by W.K.Wong@all [3,4] replaces locations with larger cloaking regions that are meant to prevent disclosure of exact user whereabouts. Nevertheless, the LBS can still derive sensitive information from the cloaked regions, so another line of research that uses cryptographic-strength protection was started by G.Ghinita @all [7] and continued in [8]. The main idea is to extend existing Private Information Retrieval (PIR) protocols for binary sets to the spatial domain, and to allow the LBS to return NN to users without learning any information about users' locations. This method serves its purpose well, but it assumes that the actual data points (i.e., the points of interest) are available in plaintext to the LBS. This model is only suitable for general-interest applications such as Google Maps, where the landmarks on the map represent public information, but cannot handle scenarios where the data points must be protected from the LBS itself. More recently, a new model for data sharing emerged, where various entities generate or collect datasets of POI that cover certain niche areas of interest, such as specific segments of arts, entertainment, travel etc. For

Published on April 2015

instance, there are social media channels that focus on specific travel habits, e.g., eco-tourism, experimental theater productions or underground music genres. The content generated is often geo-tagged, for instance related to upcoming artistic events, shows, travel destinations, etc. However, the owners of such databases are likely to be small organizations, or even individuals, and not have the ability to host their own query processing services. This category of data owners can benefit greatly from outsourcing their search services to a cloud service provider. In addition, such services could also be offered as plug-in components within social media engines operated by large industry players. Due to the specificity of such data, collecting and maintaining such information is an expensive process, and furthermore, some of the data may be sensitive in nature. For instance, certain activist groups may not want to release their events to the general public, due to concerns that big corporations or oppressive governments may intervene and compromise their activities. Similarly, some groups may prefer to keep their geo-tagged datasets confidential, and only accessible to trusted subscribed users, for the fear of backlash from more conservative population groups. It is therefore important to protect the data from the cloud service provider. In addition, due to financial considerations on behalf of the data owner, subscribing users will be billed for the service based on a *payper-result* model. For instance, a subscriber who asks for *k*NN results will pay for *k* items, and should not receive more than *k* results. Hence, approximate querying methods with low precision, such as existing techniques [5] that return many false positives in addition to the actual results, are not desirable. In this paper, we propose a family of techniques that allow processing of NN queries in an untrusted outsourced environment, while at the same time protecting *both* the POI and querying users' positions. Our techniques rely on *mutable order preserving encoding (mOPE)* [6], which guarantees *in distinguishability under ordered chosen plaintext attack* (*IND-OCPA*). We also provide performance optimizations to decrease the computational cost inherent to processing on encrypted data, and we consider the case of incrementally updating datasets.

Protecting location data is an important problem not only in the scenario of outsourced search services, but in a variety of other settings as well. For instance, two approaches for location protection have been investigated in the context of private queries to location-based services (LBS). The objective here is to allow a querying user to retrieve her nearest neighbor among a set of *public* points of interest without revealing her location to the LBS. The first approach is to use *cloaking regions (CRs)* [11,12]. Most CR based solutions implement the spatial *k*-anonymity paradigm and assume a three-tier architecture where a trusted anonymizer sits between users and the LBS server and generates rectangular regions that contain at least *k* user locations. This approach is fast, but not secure in the case of outliers. The second approach uses *private information retrieval (PIR)* protocols [7]. PIR protocols allow users to retrieve an object X from a set *X*= {X 1, X2 , … , Xn } stored by a server, without the server learning the value of *i*. The work in [7] extends an existing PIR protocol for binary data to the LBS domain and proposes approximate and exact nearest neighbor protocols. The latter approach is provably secure, but it is expensive in terms of computational overhead.

Inspired by previous work in [7] that brought together encryption and geometric data structures that enable efficient NN query processing, we investigate the use of Voronoi diagrams and Delaunay triangulations [1] to solve the problem of secure outsourced *k*NN queries. We emphasize that previous work assumed that the contents of the Voronoi diagrams [1,7] is available to the cloud provider in plaintext, whereas in our case the processing is performed entirely on cipher texts, which is a far more challenging problem. Our specific contributions are:

i)We propose the VD-*k*NN method for secure NN queries which works by processing encrypted Voronoi diagrams. The method returns exact results, but it is expensive for $k>1$ and may impose a heavy load on the data owner.

ii) To address the limitations of VD-*k*NN, we introduce T*k*NN, a method that works by processing encrypted Delaunay triangulations, supports any value of *k* and decreases the load at the data owner. T*k*NN provides exact query results for $k=1$, but when $k>1$ the results it returns are only approximate. However, we show that in practice the accuracy is high.

iii) We outline a mechanism for updating encrypted Voronoi diagrams and Delaunay triangulations that allows us to deal efficiently in an incremental manner, with changing datasets.

iv) We propose performance optimizations based on spatial indexing and parallel computation to decrease the computational overhead of the proposed techniques.

v) Finally, we present an extensive experimental evaluation of the proposed techniques and their Optimizations which shows that the proposed methods scale well for large datasets and clearly outperform competitors.

III. PRELIMINARIES

We introduce essential preliminary concepts, such as system model, privacy model and an overview of the mutable order preserving encoding (mOPE) from [6] which we use as a building block in our work.

Published on April 2015

A. System Model

The system model comprises of three distinct entities: (1) Data owner (2) the outsourced cloud service provider (for short *cloud server*, or simply *server*); and (3) Client. The entities are illustrated in Fig 1. The data owner has a dataset with *n* two-dimensional points of interest, but does not have the necessary infrastructure to run and maintain a system for processing nearest-neighbor queries from a large number of users. Therefore, the data owner outsources the data storage and querying services to a cloud provider. The dataset points of interest is a valuable resource to the data owner, the storage and querying must be done in encrypted form.

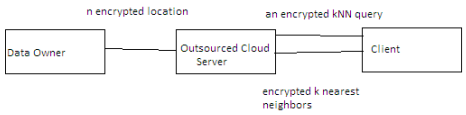


Fig. 1. System Model

The server receives the dataset of points of interest from the data owner in encrypted format, together with some additional encrypted data structures (e.g., Voronoi diagrams, Delaunay triangulations) needed for query processing. The server receives *k*NN requests from the clients, processes them and returns the results. Although the cloud provider typically possesses powerful computational resources, processing on encrypted data incurs a significant processing overhead, so performance considerations at the cloud server represent an important concern. The client has a query point *Q* and wishes to find the point's nearest neighbors. The client sends its encrypted location query to the server, and receives *k* nearest neighbors as a result. Note that, due to the fact that the data points are encrypted, the client also needs to perform a small part in the query processing itself, by assisting with certain steps.

B. Privacy Model

As mentioned previously, the dataset of points of interest represents an important asset for the data owner, and an important source of revenue. Therefore, the coordinates of the points should not be known to the server. We assume an *honest-but-curious* cloud service provider. In this model, the server executes correctly the given protocol for processing *k*NN queries, but will also try to infer the location of the data points. It is thus necessary to encrypt all information stored and processed at the server. To allow query evaluation, a special type of encryption that allows processing on cipher texts is necessary. In our case, we use the mOPE technique from [6]. mOPE is a provably secure order-preserving encryption method, and our techniques inherit the IND-OCPA security guarantee against the honest-but-curious server provided by mOPE. Furthermore, we assume that there is no collusion between the clients and server, and the clients will not disclose to the server the encryption keys.

C. Secure Range Query Processing Method

Processing kNN queries on encrypted data requires complex operations, but at the core of these operations sits a relatively simple scheme called *mutable order-preserving encryption* (*mOPE*) [6]. mOPE allows secure evaluation of range queries, and is the only provably secure order-preserving encoding system (OPES) known to date. The difference between mOPE and previous OPES techniques (e.g., Boldyreva et. al. [11,12]) is that it allows ciphertexts to change value over time, hence the *mutable* attribute. Without mutability, it is shown in [6] that secure OPES is not possible. Since our methods use both mOPE and conventional symmetric encryption (AES), to avoid confusion we will further refer to mOPE operations on plaintext/ciphertexts as encoding and decoding, whereas AES operations are denoted as encryption/decryption.

The mOPE scheme in a client-server setting works as follows: the client has the secret key of a symmetric cryptographic scheme, e.g., AES and wants to store the dataset of cipher texts at the server in increasing order of corresponding plaintexts.

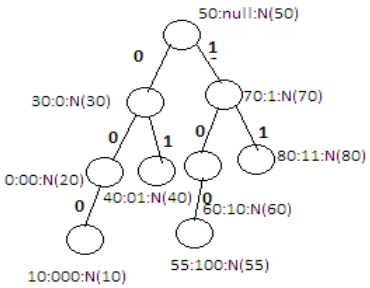


Fig 2. mOPE Tree: Inserting node *N(55)*

The client engages with the server in a protocol that builds a B-tree at the server. The server only sees the AES cipher texts, but is guided by the client in building the tree structure. The algorithm starts with the client storing the first value, which becomes the tree root. Every new value stored at the server is accompanied by an insertion in the B-tree. Figure 2 shows an example where plaintext values are also illustrated for clarity, although they are not known to the server (for simplicity we show a binary tree in the example).

mOPE encoding = [mOPE tree path]10…0

Ciphertext	mOPE Encoding
N(50)	[]1000=8
N(30)	[0]100=4

N(70)	[1]100=12
N(20)	[00]10=2
N(40)	[01]10=6
N(60)	[10]10=10
N(80)	[11]10=14
N(10)	[000]1=1
N(55)	[100]1=9

Fig.3 mOPE Table

The server maintains a mOPE table with the mapping from cipher texts to encodings, as illustrated in Fig.3 for a tree with four levels (four-bit encoding). Clearly, mOPE is an order preserving encoding, and it can be used to answer securely range queries without need to decrypt cipher texts.

IV. ONE NEAREST NEIGHBOR (1NN)

A. Voronoi Diagram-based 1NN (VD-1NN)

In this section, we focus on securely finding the 1NN of a query point. We employ Voronoi diagrams [1], which are data structures especially designed to support NN queries. An example of Voronoi diagram is shown in Figure 4. Denote the Euclidean distance between two points p and q by (p, q), and let P = {p1 , $p2$, ….pn} be a set of n distinct points in the plane. The Voronoi diagram (or tessellation) of P is defined as the subdivision of the plane into n convex polygonal regions (called *cells*) such that a point q lies in the cell corresponding to a point pi if and only if pi is the 1NN of q, i.e., for any other point pj it holds that $dist$(q, pi) <dist(q, pj) [1]. Answering a 1NN query boils down to checking which Voronoi cell contains the query point. In our system model, both the data points and the query must be encrypted. Therefore, we need to check the enclosure of a point within a Voronoi cell securely. Next, we propose such a secure enclosure evaluation scheme.

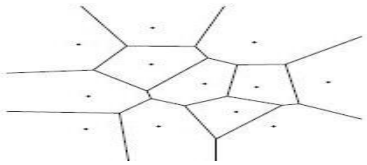


Fig.4 Voronoi Diagram

B. Secure Voronoi Cell Enclosure Evaluation

Based on the secure range query processing method , we develop a secure scheme that determines whether a Voronoi cell contains the encrypted query point. Consider the sample Voronoi cell For simplicity, we consider a triangle, but the protocol we devise works for any convex polygon as a cell. The data owner sends to the server the encrypted vertices of the cell: $V1$ (x1, y1), $V2$ (x2 , y2) and $V3$ (x3 , y3).

C. Performance Analysis

The Data Owner computes the order-1 Voronoi diagram of the data set, determines the MBR boundaries of each Voronoi cell and encodes using mOPE the cell vertices' coordinates, as well as the right side, for each edge of a Voronoi cell. The slopes, are encrypted using symmetric encryption (e.g., AES). Generation time for the Voronoi diagram is (nlogn) using Fortune's algorithm [1]. The number of Voronoi vertices that require mOPE encoding in a set of n data points is at most 2n − 5 [1]. Thus, the time to encode Voronoi points is proportional to 4n since each Voronoi point has a x-coordinate and a y-coordinate. Furthermore, the right side, must be encoded for each edge. The number of edges in a Voronoi diagram is at most 3n− 6. The total number of mOPE encoding operations is proportional. The slopes, are encrypted using AES encryption and do not require mOPE encoding. In total, the Data Owner performs 3*n* AES encryption and 7*n* mOPE encoding operations.

V. K NEAREST NEIGHBOR (KNN)

To support secure *k*NN queries, where *k* is fixed for all querying users, we could extend the VD-1NN method,by generating order-*k* Voronoi diagrams.[1]. However, this method, which we call VD-*k*NN, has several serious drawbacks:

1. The complexity of generating order-*k* Voronoi diagrams is either (k nlogn) [3] or ((n − k)o gn + nlogn) [4], depending on the approach used. This is significantly higher than (n·logn) for order-1 Voronoi diagrams.
2. The number of Voronoi cells in an order-*k* Voronoi diagram is ((n-k)), or roughly kn when $k<<n$. That leads to high data encryption overhead at the data owner, as well as prohibitively high query processing time at the server (a *k*-fold increase compared to VD-1NN).

Motivated by these limitations of VD-*k*NN, we first introduce a secure distance comparison method (SDCM) . We devise Basic *k*NN (B*k*NN), a protocol that uses SDCM as building block, and answers *k*NN queries using repetitive comparisons among pairs of data points. B*k*NN is just an auxiliary scheme, very expensive in itself, but it represents the starting point for Triangulation *k*NN (T*k*NN). T*k*NN builds on the B*k*NN concept and returns exact results for k=1. For k>1, it is an approximate method that provides high-precision *k*NN results with significantly lower costs.

VI. MATHEMATICAL MODEL

Consider two given encrypted data points Pi and Pj and encrypted query point $Q(x\text{q} , y\text{q})$. If we can securely test which data point is closer to the query point, then by repeatedly applying this test we can find all k nearest neighbors of Q.

We showed how to deter-mine whether the query point is below or above an edge of a Voronoi cell. SDCM is an extension of that scheme. where there are two data points and one query point.

First, the data owner computes the middle point of the segment that connects the two data points, denoted by p i,j, as well as the perpendicular bisector L i,j of the segment. The slope of the bisector is denoted by Si ,j . The bisector equation is:

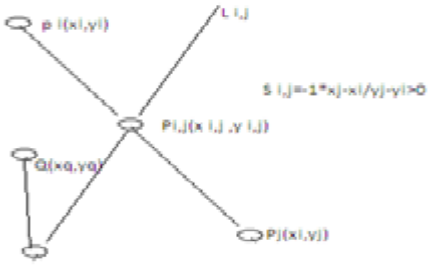
$y = -1 * (x2 - x1)/(y2 - y1) * (x - x\text{i} ,\text{j}) + y\ \text{i,j} \Leftrightarrow$
$y = S\ \text{i,j} * (x - x\ \text{i,j}) + y\ \text{i,j}$ (1)

When we plug xq into the equation, it follows that the query point is in the upper side of the bisector, hence Pi is closer to Q than Pj , if and only if

$y\ \text{q} > S\ \text{i,j} * (x\text{q} - x\ \text{i,j}) + y\ \text{i,j} \Leftrightarrow L\ \text{i,j} = y\text{q} - S\ \text{i,j} * x\text{q} > -1 * S\ \text{i,j} * x\ \text{i} ,\text{j} + y\ \text{i,j} = R\ \text{i,j}$ (2)

we observe that the right-hand side R , of Eq. (2) is independent of the query point, whereas the left-hand side L i,j depends on the query point.

The data owner can thus encode the right-hand side and send it to the server, together with the slope S i,j of the bisector. Recall that, the slope may be encrypted using conventional encryption.



VII.RESULT

Finally as a result of consideration tentatively will define a system in such a way that it works in real time and shall give a better performance and measurable scalability factors. As it works under observation of applying system on various collection of databases which inherits complex, diverse, heterogeneous and generated by autonomous sources from network.

VIII.CONCLUSION

This work revisits the secure k nearest neighbor query processing. We design two schemes first VD-Knn which based on Voronoi diagram and next is TkNN which is based on Delaunay triangulations. VD-kNN and TkNN use mutable order preserving encoding (mOPE) as building block. VD-kNN provides exact results and its performance overhead may be high. TkNN only offers approximate NN results with better performance. In addition, the accuracy of TkNN is very close to that of the exact method. In future work, we plan to investigate more complex secure evaluation functions on cipher texts, such as skyline queries. We will also research formal security protection guarantees against the client, to prevent it from learning anything other than the received k query results.

ACKNOWLEDGMENT

All faith and honor to Lord Shri Ganesh for his grace and inspiration. I wish to express my sincere thanks to Prof. Gumaste S.V. Head of Computer Engineering department and the departmental staff members for their support. Last but not the least; I would like to thank all my Friends and Family members who have always been there to support and helped me to complete this paper work in time.

REFERENCES

[1] Sunoh Choi,Gabriel Ghinita.,"Secure kNN Query Processing in Untrusted Cloud Enviroments.",IEEE PP,14
[2] Mark de Berg et.al., Computational Geometry, Springer
[3] W. K. Wong, David W. Cheung, Ben Kao, and Nikos Mamoulis, "Secure Knn Computation on Encrypted Databases.", SIGMOD'09"
[4] Haibo Hu, Jianliang Xu, Chushi Ren, and Byron Choi, "Processing Private Queries over Untrusted Data Cloud through Privacy Homomorphism",
ICDE'11
[5] Huiqi Xu, Shumin Guo, and Keke Chen,"Building Con_dential and E_cient Query Services in the Cloud with RASP Data Perturbation", TKDE'12
[6] Raluca Ada Popa, Frank H. Li, and Nickolai Zeldovich, "An Ideal-Security Protocol for Order-Preserving Encoding.", IEEE SP'13
[7] Gabriel Ghinita, Panos Kalnis, Murat Kantarcioglu, and Elisa Bertino," A Hybrid Technique for Private Location-Based Queries with Database Protection", SSTD'09
[8] Gabriel Ghinita, Panos Kalnis, Murat
Kantarcioglu, and Elisa Bertino, "Approxi-
mate and exact hybrid algorithms for private nearest-neighbor queries with database protection.", Geoinformatica'11
[9] Ali Khoshgozaran and Cyrus Shahabi, "Blind Evaluation of Nearest Neighbor Queries Using Space Transformation to Preserve Location Privacy., SSTD'07
[10] A. Boldyreva, N. Chenette, Y. Lee, and A. O'Neill, "Order Preserving Symmetric Encryption", EuroCrypt'09
[11] Jon Louis Bentley, "Multidimensional Binary Search Trees used for Associative Searching.", ACM Communications,1975
[12] Kalnis P., Ghinita G., Mouratidis K., and Papadias D., "Preserving location-based identity inference in anonymous spatial queries.,TKDE'07
28
[13] R. Agrawal, J. Kiernan. R. Srikant, and Y. Xu, "Order preserving encryption for numeric data., SIGMOD'04

[14] Der-Tsai Lee, "On k-Nearest Neighbor Voronoi Diagrams in the Plane.", IEEE Transactions on Computers, 1982
[15] Pankaj K. Agarwal, Mark De Berg, Jiri Matousek, and Otfried Schwarzkopf, "Constructing Levels in Arrangements and Higher Order Voronoi Diagrams", SIAM J.COMPUT. 1998