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

Secure kNN Query Processing in Entrusted Cloud Environments

Devidas S. Thosar[1],[2]Rajashree R Shinde[2], Prashant J. Gadakh,[3] Pratibha V. Kashid[4].
Department of Computer Engineering, SVIT,Sinner[1,3]
, Department of Electronics & Telecommunication Engineering, SVIT,Sinner[2]
SPCOE,Otur[3],
Department of Information Technology Engineering, SVIT, Sinner[4],
Pune, University ,India[1,2,3,4].
devidas.thosar@rediffmail.com[1], rajashreeshinde177@gmail.com[2], prashantgadakh31@gmail.com3, wajepratibha23@gmail.com[4]

Abstract: - Now days a Wireless devices which having geo-positioning facility like GPS enable users to give information about their current location. Users are interested in querying in their physical location like restaurants, college, home, etc. Such data may be important due to their information. Furthermore, storing such relevant information regularly to the users tedious task, so the author of such information will make the data access only to paying users. The users are send their proper location as the query parameter, and wish to accept as result the nearest position, i.e., nearest-neighbors (NNs). But actual data owners do not have the technical knowledge to support processed query on a large data, so they outsource information storage and querying to a main dataset. Many such cloud providers exist offer powerful storage and computational structures at less cost. However, such a dataset providers are not completely trusted, and typically behave in a causal fashion. Specifically they use the some rules to answer queries perfectly, but they also collect the locations of the users and the subscribers for other uses. Giving this information of locations can lead to security breaches and financial losses to the data provider, for whom the dataset is an important source of revenue. The importance of user locations leads to privacy and may refer subscribers from using the service altogether. In this paper, we propose a set of ideas that allow NN queries in an unsecured outsourced structure, while at the same time provide security to both the location and querying users' positions. Our ideas focus on only secure order-preserving encryption method which is known to-date. We also provide performance measurements to reduce the processing cost inherent to processing on secured data, and we consider the problem of incrementally updating these datasets. We present an extensive performance measurement of our ideas to illustrate their use in practice.
Keywords- location privacy, spatial databases, database outsourcing, mutable order preserving encoding.

Keywords: Data base, NN, LBS

I. INTRODUCTION

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). 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 NNsto 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 in [16, 17, 18, 19] 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 in [7] and continued in [8,9]. 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 the 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 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. Because of specificity of such dataset, collecting and maintaining such information is an tedious task, and some of the data may be important in nature. For reference, 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 important to secure this data from the dataset provider. While due to economical view on behalf of the data vendor, subscribing users will be charged for the facility based on a *paper-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 such as existing techniques [5] that gives many false results into the actual results, are not proper. In this paper, we propose a more ideas that allow processing of NN queries in an unauthorized outsourced structure, while at the same time secured *both* the location and querying users' positions. Our ides worked on *mutable order preserving encoding (mOPE)* [6], which guarantees *in distinguish ability under ordered chosen plain text attack* (*IND-OCPA*) [11,12]. We also provide performance optimizations to reduce the execution cost related to processing on secured data, and consider the case of continuously updating datasets.

Inspired by previous work in [7, 9] 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 resolve these problem of protect outsourced *k*NN queries. We propose that previous work considered that the contents of the Voronoi graph [7, 9] is available to the dataset 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 Voronoidia grams. 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.

II.RELATED WORK

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)* [16-19]. 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, 9]. PIR protocols allow users to retrieve an object X from a set $X= \{X , X , \dots , X \}$ stored by a server, without the server learning the value of *i*. The work in [7, 9] 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.

III.PRELIMINARIES

In this section, we introduce essential preliminary concepts, such as system model (Section 3.1), privacy model (Section 3.2) and an overview of the mutable order preserving encoding (mOPE) from [6] which we use as a building block in our work (Section 3.3).

A. System Model

The system model comprises of three distinct entities:(1) the data owner; (2) the outsourced cloud service provider (for short *cloud server*, or simply *server*); and (3) the client. The entities are illustrated in Figure 3-1. 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. As the dataset of points of interest is a valuable resource to the data owner, the storage and querying must be done in encrypted form (more details will be provided in the privacy model description, Section 3.2).

Figure 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., Voronoidia grams, Delaunay triangulations) needed for query processing (we will provide details about these structures in Sections 4 and 5). 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 (details will be provided in Sections 4 and 5).

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[3]. 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

As we will show later in Sections 4 and 5, 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 cipher texts 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/cipher text encoding and decoding, whereas AES operations are denoted as encryption/decryption[3].

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.

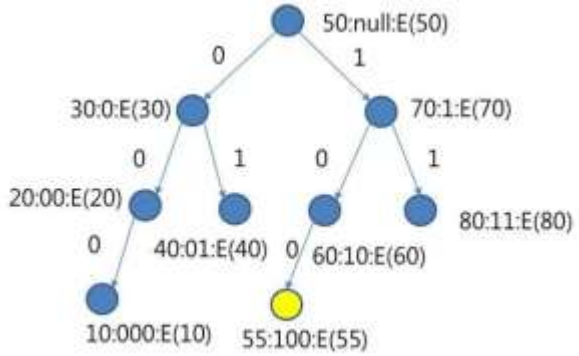

Figure 2: mOPE Tree: Inserting node *E(55)*

The client engages with theserver 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[4]. 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 3-2 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
E(50)	[]1000 = 8
E(30)	[0]100 = 4
E(70)	[1]100 = 12
E(20)	[00]10 = 2
E(40)	[01]10 = 6
E(60)	[10]10 = 10
E(80)	[11]10 = 14
E(10)	[000]1 = 1
E(55)	[100]1 = 9

Figure 3: mOPE Table

The server maintains a mOPE table with the mapping from cipher texts to encodings, as illustrated in Figure 3-3for 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 ciphertexts[4].

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 4-1. Denote the Euclidean distance between two points p and q by $di(p, q)$, and let $P = \{p , p , \dots , p \}$ be a set of n distinct points in the plane. The Voronoi diagram (or tesselation) 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 $p$if and only if p is the 1NN of q, i.e., for any other point $p$it holds that $dist(q, p) < dist(q, p)$ [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[10,14]. Therefore, we need to check the enclosure of a point within a Voronoi cell securely. Next, we propose such a secure enclosure evaluation scheme[14].

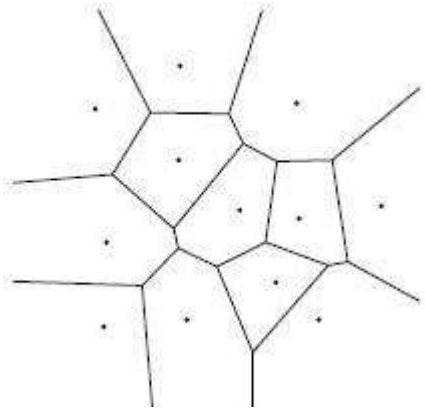

Figure 4:Voronoi Diagram

B. Secure Voronoi Cell Enclosure Evaluation

Based on the secure range query processing method introduced in Section 3.3, we develop a secure scheme that determines whether a Voronoi cell contains the encrypted query point. Consider the sample Voronoi cell from Figure 4-2. For simplicity, we consider a triangle, but the protocol we devise works for any convex polygon as a cell[10]. The data owner sends to the server the encrypted vertices of the cell: $V(,y)$, $V (x , y)$ and $V (x , y)$.

C. Performance Analysis

The Data Owner computes the order-1 Voronoi diagram of the dataset, determines the MBR boundaries of each Voronoi cell and encodes using mOPE the cell vertices' coordinates, as well as the right side ,of Eq. (9) 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 Voronoivertices that require mOPE encoding in a set of $n$data points is at most $2n - 5$ [1,14]. Thus, the time to encode Voronoipoints is proportional to $4n$since each Voronoipoint has a x-coordinate and a y-coordinate. Furthermore, the right side , of Eq. (9) 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 to 7 [14]. The slopes ,are encrypted using AES encryption and do not require mOPE encoding. In total, the Data Owner performs $3n$ AES encryption and $7n$mOPE encoding operations.

V.K NEAREST NEIGHBOR (KNN)

To support secure kNN queries, where k is fixed for all querying users, we could extend the VD-1NN method

from Section 4 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 $(knlogn)$ or $(k(n-k)logn + nlogn)$, depending on the approach used. This is significantly higher than $(n \cdot logn)$ for order-1 Voronoidiagrams.

(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)[13,14].

Motivated by these limitations of VD-*k*NN, we first introduce a secure distance comparison method (SDCM) in Section 5-1. Next, in Section 5-2 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[2]. 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), presented in Section 5-3. 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-valued *k*NN results with significantly lower budget[2].

V.CONCLUSION

In this paper, we proposed two schemes to support secure *k* nearest neighbor query processing: VD-*k*NN which is based on Voronoi diagrams, and T*k*NN which relies on Delaunay triangulations. They both use mutable order preserving encoding (mOPE) as building block. VD-*k*NN provides exact results, but its performance overhead may be high. T*k*NN only offers approximate NN results, but with better performance. In addition, the accuracy of T*k*NN is very close to that of the exact method. In future work, we plan to investigate more complex secure evaluation functions on ciphertexts, 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.

VI.REFERENCES

[1] Mark de Berg et.al., Computational Geometry, Springer
[2] W. K. Wong, David W. Cheung, Ben Kao, and Nikos Figure 8-3. Data Encryption Time Mamoulis, Secure kNN Computation on Encrypted Databases, SIGMOD'09
[3] Haibo Hu, JianliangXu, ChushiRen, and Byron Choi, Processing Private Queries over Untrusted Data Cloud through Privacy Homomorphism, ICDE'11
[4] HuiqiXu, ShuminGuo, and Keke Chen, Building Confidential and Efficient Query Services in the Cloud withRASP Data Perturbation, TKDE'12
[5] Bin Yao, Feifei Li, and Xiaokui Xiao, Secure Nearest Neighbor Revisited, ICDE'13
[6] Raluca Ada Popa, Frank H. Li, and NickolaiZeldovich, An Ideal-Security Protocol for Order-Preserving Encoding, IEEE S&P'13
[7] Gabriel Ghinita, PanosKalnis, Ali Khoshgozaran, Cyrus Shahabi, and Kian-Lee Tan, Private Queries in Location Based Services: Anonymizers are not Necessary, SIGMOD' 08
[8] Gabriel Ghinita, PanosKalnis, Murat Kantarcioglu, and Elisa Bertino, A Hybrid Technique for Private Location-Based Queries with Database Protection, SSTD'09
[9] Gabriel Ghinita, PanosKalnis, MuratKantarcioglu, and Elisa Bertino, Approximate and exact hybrid algorithms for private nearest-neighbor queries with database protection, Geoinformatica'11
[10] Ali Khoshgozaran and Cyrus Shahabi, Blind Evaluation of Nearest Neighbor Queries Using Space Transformation to Preserve Location Privacy, SSTD'07
[11] A. Boldyreva, N. Chenette, Y. Lee, and A. O'Neill, Order Preserving Symmetric Encryption, EuroCrypt'09
[12] A. Boldyreva, N. Chenette, and A. O'Neill, Order_preserving Encryption Revisited: Improved Security Analysis and Alternative Solutions, Crypto'11
[13] Jon Louis Bentley, Multidimensional Binary Search Trees used for Associative Searching, ACM Communications, 1975
[14] Thomas Roos, Voronoi diagrams over dynamic scenes, Discrete Applied Mathematics, 1993.
[15] Gruteser M. and Grunwald D., Anonymous usage of location-based services through spatial and temporal cloaking, MOBISYS'03
[16] Gedik B. and Liu L., Location privacy in mobile systems: a personalized anonymization model, ICDCS'05
[17] Mokbel M. F., Chow C. Y., and Aref W. G., The new Casper: query processing for location services without compromising privacy, VLDB'06
[18] Kalnis P., Ghinita G., Mouratidis K., and Papadias D., Preserving location-based identity inference in anonymous spatial queries, TKDE'07
[19] R. Agrawal, J. Kiernan. R. Srikant, and Y. Xu, Order preserving encryption for numeric data, SIGMOD'04