2017 IEEE 33rd International Conference on Data Engineering

Secure KNN Queries over Encrypted Data: Dimensionality is not Always a Curse

Xinyu Lei* Alex X. Liu* Rui Li†
*Department of Computer Science and Engineering, Michigan State University
†College of Computer Science and Network Security, Dongguan University of Technology
Email: {leixinyu, alexliu}@cse.msu.edu, ruili@dgut.edu.cn

***Abstract*—The fast increasing location-dependent applications in mobile devices are manufacturing a plethora of geospatial data. Outsourcing geospatial data storage to a powerful cloud is an economical approach. However, safeguarding data users' location privacy against the untrusted cloud while providing efficient location-aware query processing over encrypted data are in conflict with each other. As a step to reconcile such conflict, we study secure k nearest neighbor (SkNN) queries processing over encrypted geospatial data in cloud computing. We design 2D SkNN (2DSkNN), a scheme achieves both strong provable security and high-efficiency. Our approach employs *locality sensitive hashing* (LSH) in a *dimensional-increased* manner. This is a counter-intuitive leverage of LSH since the traditional usage of LSH is to reduce the data dimensionality and solve the so-called "curse of dimensionality" problem. We show that increasing the data dimensionality via LSH is indeed helpful to tackle 2DSkNN problem. By LSH-based neighbor region encoding and *two-tier prefix-free encoding*, we turn the proximity test to be sequential keywords query with a stop condition, which can be well addressed by any existing *symmetric searchable encryption* (SSE) scheme. We show that 2DSkNN achieves adaptive indistinguishability under chosen-keyword attack (IND2-CKA) secure in the random oracle model. A prototype implementation and experiments on both real-world and synthetic datasets confirm the high practicality of 2DSkNN.**

I. Introduction

With the proliferation of location-based services on clouds, protecting the privacy of data while keeping data utility is of great importance. Most of the mobile devices (like smartphones and mobile vehicles) nowadays are equipped with GPS and many mobile applications (such as Google Map and Facebook) provide location-based services by sending the current user location as a geospatial query parameter to a remote cloud and the cloud returns the corresponding query results (such as the list of nearby restaurants). By resorting to the cloud storage, data owners can gain tremendous economic savings and the data users can enjoy the convenience of location-based services. Many commercial companies, such as Dropbox Inc., Microsoft Inc., are providing free or cheap storage capacity on servers they administer. Despite the tremendous benefit of cloud storage, the security concern is the key road block for its development due to the fact that the public clouds are not fully trusted. On one hand, data owners are concerned with the privacy of their data, which is outsourced to the cloud. On the other hand, data users are concerned with their location privacy because their locations are leaked to the cloud in the query processing process. There are strong financial incentives for the public cloud to collect its customers' sensitive information. The cloud may sell the collected information for money. Moreover, public clouds may be hacked and the stored information may be leaked. For example, a recent report [1] shows that dropbox has been hacked and more than 68 million account details are now for sale on the darknet marketplace. Therefore, it is crucial to provide privacy preserving countermeasures for location-based queries in cloud storage.

The type of query we study in this work is secure k nearest neighbor (SkNN) query, which is a very popular geospatial data query. For instance, taxi drivers may often want to retrieve top five nearest restaurants. Therefore, we aim to design a scheme that provides strong data privacy against the untrusted cloud storage server, while still reserves the cloud's ability to efficiently answer kNN queries over encrypted geospatial data.

A. Problem Formulation

Relaxed kNN. The kNN query processing problem we consider can be mathematically described as follows: suppose that there are a set of spatial data items that are represented by points $p_1, \cdots, p_n$ in the two-dimensional (2D) geographical work space U, given a query point $q \in U$, the target of kNN is to find top-k nearest points of q. The distance metric we use is the Euclidean distance. It is shown that an approximate answer of kNN suffices for many applications, so we do not insist on the exact answer. Accordingly, the problem we attempt to tackle is a relaxation of kNN problem. Such granularity of relaxation well captures most location-based practical applications, while offering more space for protocol design.

System Model. We consider 2D SkNN query processing system model as depicted in Fig. 1, where there exist a data owner and a cloud. We adopt the *index-aid cryptographic approach* to solve SkNN problem. In this approach, data owner and authenticated data users share some secret keys in advance. Each data item hosted by the data owner consists of location information (*spatial attributes*) and other information (*non-spatial attributes*). In order to remain the ability to query and retrieve the data efficiently, the data owner extracts the spatial attributes of each data item and builds a secure index and then encrypts the entire data item by using the shared keys. Each secure index item should contain the identifier information to record the association between the secure index item and the encrypted data item. Afterward, the data owner

2375-026X/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDE.2017.91

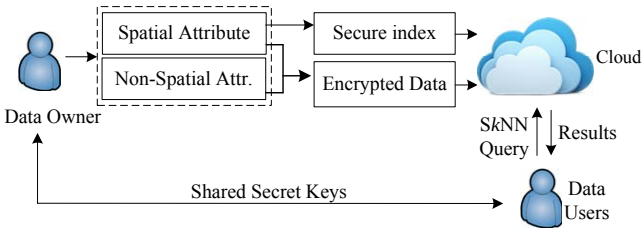

Fig. 1. 2D SkNN System Model

outsources both the secure index items and the encrypted data items to the powerful cloud, which provides both storage and search services. After the cloud receives the secure index and encrypted data items, the authorized data users can use the shared keys to generate valid search tokens and search for the corresponding SkNN results.

Threat Model. In the system model, the security threats primarily come from the behavior of the public cloud. We assume that the cloud is *semi-honest*, or equivalently, *honest-but-curious*. In the semi-honest threat model, the cloud correctly follows the protocol specification. However, the cloud records all the information it can access and attempts to use this to learn information that should remain private.

Design Goals. Security. Informally speaking, the protocol should satisfy three-fold. 1) *Ciphertext privacy*: the encrypted data items reveal no useful information about the data. 2) *Index privacy*: from the encrypted index, the adversary cannot learn any useful information about the spatial information of the data items. 3) *Token privacy*: from the encrypted search token, the adversary cannot infer any information about the query point's location. Result Accuracy. The results obtained are of high accuracy in comparison with the ground truth. Efficiency. Two aspects regarding efficiency should be satisfied. 1) *Low query latency*: the data user can get the result within a reasonable amount of time even if the data user is of low computation ability and there are a huge amount of points in the dataset. 2) *Low communication overhead*: first, the data transferred between the data owners and the cloud, other than the encrypted data items, is low. Second, except the query results, the amount of data exchanged between the data users and the cloud should be small. Third, the protocol should be *non-interactive*, or it just requires a small constant number of interactions.

B. Limitation of Prior Art

One of the major limitation of previous solutions that are applicable to solve 2D SkNN is that while some solutions can achieve strong provable security, but they are not sufficiently efficient, such as *non-index cryptographic approach* (e.g., [2]–[5]). At a high level, the intuition why non-index cryptographic approaches suffer from inefficiency is: if the strong *semantic secure* encryption algorithm is employed, namely the ciphertext does not leak any information to the adversary, then every encrypted data item needs to be touched in order to reveal information inside it. As a result, the optimal query processing time of non-index semantic secure cryptographic solutions is linear, which is still prohibitively expensive for large dataset. The other major limitation is that while some solutions can achieve high-efficiency, but they suffers from weak security guarantee, such as *location obfuscation approach* (e.g., [6], [7]) and *data transformation approach* (e.g., [2], [8]). These approaches do not employ formal encryption methods, so it is not easy for them to achieve provable security. In summary, prior works suffer from either weak security guarantee or insufficient efficiency.

C. Proposed Approach

In order to obtain both high-efficiency and strong provable security guarantee, we adopt the index-aid cryptographic approach to cope with the 2D SkNN problem. In this approach, the data owner extracts the spatial attributes of each data record to build a secure index and encrypts the entire records via a formal encryption algorithm. Then, the data owner outsources both the encrypted data and secure index to the cloud, which provides storage and search services. On one hand, by building index before outsourcing, the high-efficiency can be achieved. One the other hand, by formal encryption of the data items, the strong security can be ensured.

D. Technical Challenges and Solutions

The first technical challenge is how to achieve efficient sublinear search time. Our solution is first to perform implicit space encoding by employing locality sensitive hashing (LSH) in a *dimension-increased* manner. The LSH codes can help us to transform the secure kNN computational problem (i.e., computation over ciphertext) to be a decisional problem with an only yes-or-no answer (i.e., `Is_exist` evaluation). Then, through organizing neighbor region codes into a binary tree structure, the cloud is able to directly pinpoint a near neighbor region and search the points therein. This can help the cloud to prune most of the points in the space and perform *incremental search* from the smallest neighbor region of the queried point to the largest one and stop once enough points are found.

The second technical challenge is how to use LSH to realize *proximity test* over 2D space. The traditional LSH is designed for high-dimensional data, there are no existing LSH that works directly for 2D data. Therefore, we first design a special 2DLSH that works over 2D space by a slight generalization of the traditional LSH. We then use single 2DLSH to encode an infinite space and use 2DLSH composition to encode a finite neighbor region. LSH-based neighbor region encodes enable us to perform proximity test by equality checking. We want to emphasize that the idea of employing 2DLSH to increase the data dimensionality to better differentiate near points is not unique. For example, the kernel support vector machine (SVM) projects data to high-dimensional space in order to better classify them.

The third technical challenge is how to improve the result accuracy. In order to achieve the high result accuracy, we identify the *successive inclusion property*, which means that by carefully adjusting parameters in 2DSkNN, a series of

successive gradually increased near neighbor regions can be obtained. This property allows the cloud to perform *repeated filtering* for the points in the nearest neighbor. Therefore, the proposed 2DSkNN scheme can achieve *top nearest accuracy property*, by which we mean that the nearer the point, the less probability it will be missed in the search process. This property is of practical significance since most of the decisions we made (e.g., which restaurant to choose) are always among the top nearest query results.

II. Space Encoding

The traditional LSH is designed for high-dimensional data, there are no existing LSH that works for 2D data. In this section, we first construct a specific LSH that works over 2D space and then illustrate how to encode space by a single 2DLSH and 2DLSH composition, respectively.

A. 2DLSH Introduction

We design the following conceptually simpler and elegant LSH (named 2DLSH) that works over 2D space.

(2DLSH) 2DLSH is defined as

$$h(q) = \lfloor \frac{\vec{a} \cdot \vec{q} + b}{d} \rfloor,$$

where $\vec{a} = (\theta, r)$ denotes the vector in polar coordinate form, the angle $\theta \leftarrow U[0, 2\pi)$ and the radius $r = 1$. b is a random variable follows $b \leftarrow [0, d)$.

To facilitate description, we use $\mathcal{H}$ to denote the 2DLSH family that contains all of the 2DLSHs generated by Eq. (II-A). Likewise, we denote by $g \leftarrow \mathcal{H}$ the process of randomly sampling a 2DLSH g from 2DLSH family $\mathcal{H}$.

B. 2DLSH Space Encoding

It is observed that 2DLSH can be used to encode a geometric region. We now explain such observation by casting a micro-view of the geometric interpretation of a single 2DLSH.

Given a point $q \in \mathbb{R}^2$, let's investigate the *feasible region* of p such that $h(p) = h(q)$, where $h \leftarrow \mathcal{H}$. The property of the feasible region is identified by the following Theorem.

Theorem II.1. *Given a point q, the feasible region of p such that $h(p) = h(q)$ is between two parallel lines l_1 and l_2 that are perpendicular to $\vec{a}$. Define the width of the feasible region wid as the distance of l_1 and l_2, we have $wid = d$, which is independent of the location of p and the choice of b.*

According to Theorem II.1, 2DLSH code value enables us to test whether two points p, q are in the same infinite d-width space by checking $h(q) \stackrel{?}{=} h(p)$.

C. 2DLSH Composition Space Encoding

It has been shown that a single 2DLSH allows us to encode an *infinite space*, we now continue to illustrate how to use the 2DLSH composition to encode a *finite space*. Given a 2DLSH family $\mathcal{H}$, new LSH families can be constructed by either And-composition or Or-composition.

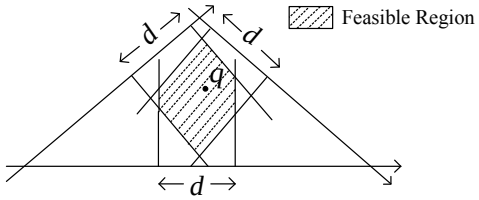

Fig. 2. Illustration of Feasible Region by Using 2DLSH And-composition

Definition II.1. *1) (And-composition) A new family $\mathcal{H}^d_{v,1}$ is defined by taking v pairwise independent functions $h_1, \cdots, h_v$ with interval length d from $\mathcal{H}$. For a hash function $g \in \mathcal{H}^d_{v,1}$, $g(p) = g(q)$ iff $h_i(p) = h_i(q)$ for all $i \in [v]$, where $[v]$ denotes the set $\{1, \cdots, v\}$.*

2) (Or-composition) A new family $\mathcal{H}^d_{1,t}$ is defined by taking t pairwise independent functions $h_1, \cdots, h_t$ with interval length d from $\mathcal{H}$. For a hash function $g \in \mathcal{H}^d_{1,t}$, $g(p) = g(q)$ iff $h_i(p) = h_i(q)$ for at least one $i \in [t]$, where $[t]$ denotes the set $\{1, \cdots, t\}$.

We use $\mathcal{H}^d_{v,t}$ to represent LSH family as constructed by the following two steps. Step 1): we first apply And-composition of v pairwise independent 2DLSH with interval length d from $\mathcal{H}$ and obtain LSH family $\mathcal{H}^d_{v,1}$. Step 2): we continue to apply Or-composition of t pairwise independent LSH from $\mathcal{H}^d_{v,1}$ and obtain LSH family $\mathcal{H}^d_{v,t}$.

Given a point q, we now study the feasible region of p such that $g(p) = g(q)$, where $g \leftarrow \mathcal{H}^d_{v,1}$. Taking $v = 3$ as an example, the geometric illustration of the feasible region is shown in Fig. 2. The LSH g specifies three randomly chosen vectors. According to Theorem II.1, the feasible region of p with respect to each vector is between a pair of parallel lines with width d. Since g is constructed by three And-composition, the feasible region is the *intersected area* between three pairs of parallel lines, i.e., the *polygon region* shown in Fig. 2. With a slight abuse of terminology, we denote the feasible region specified by g ($g \leftarrow \mathcal{H}^d_{v,1}$), as a $\mathcal{H}^d_{v,1}$-neighbor-region ($\mathcal{H}^d_{v,1}$-NR, for short) of q.

We continue to study the feasible region by incorporating Or-composition of LSH. Given a point q, where is the feasible region of p such that $g(p) = g(q)$ ($g \leftarrow \mathcal{H}^d_{v,t}$)? According the definition of Or-composition, it not hard to know that the feasible region is the *union* of t $\mathcal{H}^d_{v,1}$-NRs of q. Likewise, the corresponding feasible region is denoted as a $\mathcal{H}^d_{v,t}$-neighbor-region ($\mathcal{H}^d_{v,t}$-NR) of q.

In analogy to the single 2DLSH space encoding, 2DLSH composition code values enable us to perform *proximity testing*. More specifically, we can test whether two points p, q are in the same finite space $\mathcal{H}^d_{v,t}$-NR of q by checking $g(p) \stackrel{?}{=} g(q)$, where $g \leftarrow \mathcal{H}^d_{v,t}$.

III. kNN Protocol

We now describe the process of kNN query processing *in plaintext domain*. Consider the system model as dipicted in Fig. 1. The system setups several global parameters including v, t, and L successive increasing interval lengths $(d_1, \cdots, d_L)$.

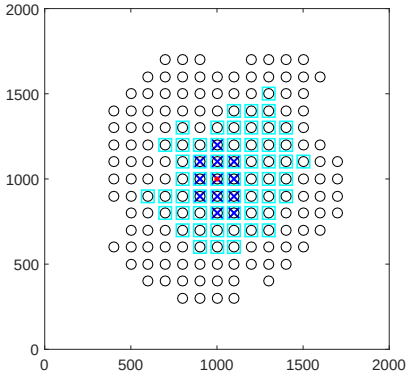

Fig. 3. Illustration of Successive Inclusion Property

The cloud hosts a dataset of n data items in plaintext, then it extracts the spatial attributes, denoted as $p_1, \cdots, p_n$. The L successive increasing interval lengthes $(d_1, \cdots, d_L)$ can result in a couple of neighbor regions of q: $\mathcal{H}_{v,t}^{d_1}$-NR,$\cdots$, $\mathcal{H}_{v,t}^{d_L}$-NR. If v and t are properly chosen and the gap between two successive values in $(d_1, \cdots, d_L)$ is sufficiently large, it almost holds that$\mathcal{H}_{v,t}^{d_1}$-NR $\subset \mathcal{H}_{v,t}^{d_2}$-NR $\subset \cdots \subset \mathcal{H}_{v,t}^{d_L}$-NR. We call this property as *successive inclusion property*, which is illustrated by an experiment as shown in Fig. 3, where cross symbol represents a $\mathcal{H}_{10,20}^{400}$-NR, square symbol represents a $\mathcal{H}_{10,20}^{800}$-NR, and circle symbol represents a $\mathcal{H}_{10,20}^{1600}$-NR. All of them are generated with respect to point q in $(1000, 1000)$, which is depicted as a big and red dot in Fig. 3.

The design rationale of the kNN protocol is verbally interpreted as follows. First, the cloud invokes Index-Building Alg. to compute and store LSH values of each point in the index. Then, the data user calls the Token-Generation Alg. to compute and store LSH values of the query point in a token array. Recall that whether two points are in the same region or not can be deduced by comparing their LSH values, the cloud calls Query-Processing Alg. to check whether $p_i (i \in [L])$ is in the smallest $\mathcal{H}_{v,t}^{d_1}$-NR with query point q via checking $g_1(q) \stackrel{?}{=} g_1(p_i)$. According to the successive inclusion property, the cloud searches from the smallest neighbor region $\mathcal{H}_{v,t}^{d_1}$-NR to the largest one $\mathcal{H}_{v,t}^{d_L}$-NR and stops until k distinct points are found. Upon receiving the k distinct points, the data user computes their distance to the query point and then sorts them to get the approximate kNN results.

IV. Methodologies for Secure and Sublinear Protocol

With the help of LSH-based space encoding, we actually turn the kNN query processing to be sequential keyword query processing with a stop condition. Consequently, we can turn our kNN protocol to be a secure and sublinear protocol by using *any existing* protocol SSE scheme. Our method minimizes the probability of designing a flawed protocol, as long as the underlying SSE scheme is secure. As a proof of concept, we make use of the recently proposed indistinguishable Bloom filter (IBF) [9] index structure, since IBF enables cloud to realize high-efficiency in query processing. See [9] for further details.

V. Experimental Results Summary

The intensive experiments on both real-world and synthetic datasets demonstrate 2DSkNN scheme is unprecedentedly fast. More specifically, the query latency for 50NN is less than 50 ms in datasets with 1 million points in our experiments. Meanwhile, 2DSkNN scheme can achieve high result accuracy, the average *overall approximation ration* (OAR) of the results is about 1.3 on our datasets. Furthermore, by employing the successive inclusion property, we can improve the average OAR to be about 1.1.

VI. Conclusion

In this demonstration, we have illustrated the design rationale of 2DkNN scheme, which enables practical SkNN query processing over encrypted geospatial data in cloud computing. Via a counter-intuitive leverage of LSH and then by employing SSE, the constructed 2DkNN scheme can simultaneously fulfill the design goals of security, result accuracy, and high-efficiency. Our study demonstrates that dimensionality is not always a curve: increasing data dimensionality via LSH is indeed helpful to tackle the 2D SkNN problem. At a high level, schemes based on LSH for high-dimensional kNN search succeed in trading off result accuracy for speed up. Analogously, 2DSkNN scheme succeeds in trading off result accuracy for both security and speed up. Therefore, the proposed 2DSkNN scheme is of great practical significance in the era when big data meets security.

Acknowledgment

This work is supported by the National Science Foundation under Grant Numbers CNS-1318563, CNS-1524698, CNS-1421407, and IIP-1632051, the National Natural Science Foundation of China under Grant Numbers 61472184, 61321491, 61370226, and 61672156, and the Jiangsu Innovation and Entrepreneurship (Shuangchuang) Program.

References

[1] "Dropbox hack," https://www.troyhunt.com/the-dropbox-hack-is-real/.
[2] W. K. Wong, D. W.-l. Cheung, B. Kao, and N. Mamoulis, "Secure knn computation on encrypted databases," in *SIGMOD*, 2009, pp. 139–152.
[3] B. Yao, F. Li, and X. Xiao, "Secure nearest neighbor revisited," in *ICDE*, 2013, pp. 733–744.
[4] H. Hu, J. Xu, C. Ren, and B. Choi, "Processing private queries over untrusted data cloud through privacy homomorphism," in *ICDE*, 2011, pp. 601–612.
[5] Y. Elmehdwi, B. K. Samanthula, and W. Jiang, "Secure k-nearest neighbor query over encrypted data in outsourced environments," in *ICDE*, 2014, pp. 664–675.
[6] M. F. Mokbel, C.-Y. Chow, and W. G. Aref, "The new casper: query processing for location services without compromising privacy," in *VLDB*, 2006, pp. 763–774.
[7] M. L. Yiu, C. S. Jensen, X. Huang, and H. Lu, "Spacetwist: Managing the trade-offs among location privacy, query performance, and query accuracy in mobile services," in *ICDE*, 2008, pp. 366–375.
[8] A. Khoshgozaran and C. Shahabi, "Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy," in *SSTD*, 2007, pp. 239–257.
[9] R. Li and A. X. Liu, "Adaptively secure conjunctive query processing over encrypted data for cloud computing," in *ICDE*, 2017.