World Wide Web (2019) 22:437–454
https://doi.org/10.1007/s11280-017-0507-4

# A resource-aware approach for authenticating privacy preserving GNN queries

**Yan Dai**[1] · **Jie Shao**[1] · **Gang Hu**[1] · **Long Guo**[2]

Received: 13 August 2017 / Revised: 22 September 2017 / Accepted: 16 October 2017 /
Published online: 6 December 2017

**Abstract** Nowadays many location service providers (LSPs) employ spatial databases outsourced from a third-party data owner (DO) to answer various users' queries, e.g., group nearest neighbor (GNN) queries that enable a group of users to find a meeting place minimizing their aggregate travel distance. Along with the benefits from LSPs and DO, protection of location privacy and authentication of query results become two major concerns for users while assessing GNN queries. This paper proposes a resource-aware approach that supports effective location privacy preservation and efficient query result authentication with a less storage, communication and computation overhead. Specifically, two centroid-based techniques are investigated to generate a centroid point, which initiates GNN query on behalf of the group members. Then, an authentication algorithm based on Voronoi diagram is proposed for spatial queries. Finally, we demonstrate how our approach is resistant to various attacks, and evaluate its performance by comparing with three competitive approaches.

This article belongs to the Topical Collection: *Special Issue on Deep vs. Shallow: Learning for Emerging Web-scale Data Computing and Applications*.
Guest Editors: Jingkuan Song, Shuqiang Jiang, Elisa Ricci, and Zi Huang

✉ Jie Shao
shaojie@uestc.edu.cn

Yan Dai
yandai@std.uestc.edu.cn

Gang Hu
hugang@std.uestc.edu.cn

Long Guo
guolong@pku.edu.cn

[1] Center for Future Media & School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China

[2] School of Electronics Engineering and Computer Science, Peking University, Beijing, China

The results show the proposed approach is better and more economical in terms of resource overhead, while considering both privacy preservation and query authentication.

**Keywords** Group nearest neighbor queries · Location privacy · Query authentication

## 1 Introduction

The popularity of mobile positioning techniques enables users to obtain various types of location-based services (LBSs) from location service providers (LSPs) for a more convenient way of life. Just like similarity search in high-dimensional space with feature vectors [28, 34], the most common forms of spatial queries are users querying their nearby neighbors or points of interest (POIs) (i.e., $k$-nearest neighbors), or querying locations of all POIs within a space area (i.e., range neighbors). Besides requests from a single user, a group of users sometimes need to find a meeting place, with the minimum aggregate travel distance, namely *group nearest neighbor* (GNN) [22]. The aggregate travel distance can be measured in terms of minimizing the total or the maximum distance from all group members to the meeting place. While LBSs have brought great convenience and commercial value, there are serious issues that need to be addressed urgently.

As various LBSs have grown at an exceptional pace, the large amount and great complexity of spatial information would demand more sophisticated data management systems. However, this is beyond the abilities of many LSPs, which possess limited processing capabilities or little technical expertise. Consequently, the idea of *database outsourcing* becomes popular, in which a data owner (DO) has been introduced for addressing this problem [33]. It is now common that DO and LSP are two different organizations. In this paradigm, the DO is responsible for storage and management of spatial data, and the LSP could index the data from the DO and then answer various user queries while only storing the most necessary spatial information. Because there is no need to store all the complex spatial information in the LSP side, the storage overhead would be greatly reduced, so we adopt the idea of DO in this paper.

Moreover, while accessing these LBSs, users are supposed to expose their exact locations in order to obtain high-quality results. However, such exposure may lead untrusted severs or malicious attackers derive sensitive and private information (such as health conditions, economic incomes, living habits and religious beliefs) through analyzing the users' locations or trajectories. Therefore, serious privacy breaches occur [5]. Generally, privacy preservation in LBS is roughly divided into two models: client-server (CS) and peer-to-peer (P2P). In the CS model, privacy preservation is achieved through effective algorithms such as using imprecise location methods of *k-anonymity* [6, 29] and *cloaking region* [6, 30], adding Gaussian noise with zero mean of *differential privacy* [1], exposing along with additional fake locations of *dummy locations* [19], using a centralized trusted third party such as *anonymizer* [24], or using cryptograph methods of *encryption protocol* such as space transformation [18] and private information retrieval techniques [8]. In the P2P model, users seek query results with a cooperative computation based on the locations of neighboring members, such as [7, 13]. Normally for the P2P model, the capabilities of user devices still cannot deal with the huge amount of complex spatial data. Thus, we adopt the CS model in this study.

In particular, processing GNN queries in a privacy preserving manner has not been fully investigated so far. Our study focuses on GNN queries in the CS model with the participation of a DO. In this paradigm, users concern about not only whether their locations would be known by untrusted parties, but also the quality of answers provided by the LSP, because the LSP may collude with a third party for profit and provide low-quality answers. That is to say, we should not only protect all group users' locations of GNN queries, but also guarantee the total distance from all group users to the obtained location of meeting palace is smallest. Therefore, our work jointly considers privacy preservation of users's locations and result authentication of spatial queries.

As for location privacy, we aim to protect against potential attackers inside or outside the group. Our approach adopts a centroid-based privacy preserving method. We generate a centroid point as the group representative according to the group members' locations. Then, a *coordinator*, who is randomly selected within the group, uses the centroid point to initiate the GNN query. Finally, the LSP returns the exact result to the coordinator through indexing data from the DO. Two techniques are extended to generate the centroid point, namely *cloaked centroid* based on geometrical cloaking region [3] and *encrypted centroid* based on an anonymous veto network (AV-net) [9]. As the LSP just receives a query request and then returns a strong result (i.e., the nearest POI of the centroid point), there is no need to refine the answer set.

As for query authentication, we propose our authentication algorithm based on neighboring spatial information derived from the Voronoi diagram of the underlying spatial index. The essence is that, the DO stores all POIs in the form of Voronoi diagram, and the information of each POI includes its location, the distance from itself to the farthest Voronoi neighbor (i.e., POI), the number of other POIs that have a smaller distance than the farthest distance, and the signed digital signatures. Then, the DO shares all POI information with the LSP. For the LSP, query processing is also based on the Voronoi diagram, and it returns the query result (i.e., POI) and additional POIs for the authentication. Then, we verify the location of the query result through four steps. The details of verification process will be given later, and our algorithm incrementally retrieves the answered POI until the correct answer is verified.

Our main contributions are summarized are as follows:

- We study the privacy preserving issue of answering GNN queries in a database outsourcing scenario, with emphasis on resource saving. Apart from effectively protecting user location privacy, it greatly reduces the communication overhead and computation cost because of fewer queries, only a strong result and more reasonable query processing.
- We propose an efficient authentication algorithm based on Voronoi diagram for spatial queries. It not only ensures the reliability and correctness of query results, but also reduces the communication cost and storage load.
- We show how our algorithm can be resistant to various attacks including *collusion attack*, *knowledge attack* and *distorting attack*, in terms of effectively preserving location privacy.
- We conduct our experiment on a synthetic dataset and compare with the existing proposals in terms of communication, computation and storage overhead. The results demonstrate that our approach outperforms these competitive approaches.

The rest of the paper is organized as follows. In Section 2, we review related studies. We provide some background knowledge and show the system architecture in Section 3. In Section 4, we introduce the proposed approach. Section 5 presents our performance evaluation and Section 6 concludes.

## 2 Related work

We discuss related studies from two aspects: privacy preservation and answer authentication.

### 2.1 Privacy preservation for GNN queries

Compared with location privacy preservation for a single query user, privacy preservation for GNN queries is more complex. Hashem et al. [10] propose a privacy preserving GNN query processing technique (called "PGNN" in the experiment) based on two phases. In the first phase, users provide their imprecise locations (cloaking regions) to the LSP. Then, the LSP returns a set of candidate POIs. In the second phase, users need to filter out the exact result mainly based on secure multi-party computations [25]. Although this method could protect the location privacy of group members, it leads to excessive computation and communication overhead.

Ashouri-Talouki et al. take a cryptographic approach for the privacy preserving GNN queries with two solutions. In [2], each member publishes a masked location, and a special member $u_a$ computes the encrypted centroid point. Afterwards, the LSP decrypts the encrypted centroid and returns the nearest POI of that centroid as the query result. This method leads to a low communication overhead and computation cost. Besides, there is no need to refine the answer set because it only contains the exact result. Unfortunately, the nearest POI of the centroid might not be the correct GNN result.

The follow-up work of [2] is reported in [3]. The proposed method (called "CCP" in the experiment) submits a single GNN query along with the cloaking region of centroid point to the LSP, and receives an answer set. Then, it privately determines the exact result. This method is resource-aware in the sense that its processing cost is low, but the filtering process is also needed. In addition, it cannot ensure the reliability and correctness of query result as well.

There are some other studies on privacy preservation of GNN queries. Huang and Vishwanathan [16] explore the application of secure multi-party computation techniques based on garbled circuits. Hashem et al. [11] focus on protecting user privacy that evaluates LBSs with crowdsourced data and computation. In addition, [17] propose a naive method, which does not require a user to provide location information, and thus is location oblivious.

### 2.2 Authentication for spatial queries

LSPs process GNN queries based on spatial information from a third-party DO. Due to the nature of cloud computing, there are many security issues in the cloud service provided by the third party [26, 27]. Many investigations have been made for authenticating query results in outsourced databases [4, 12, 14, 15, 23, 32]. However, some of them do not take a user's location privacy into account [15, 23, 32], or focus on authenticating query results and preserving privacy of the stored data in LSP's database [4, 14].

Approaches based on digital signature [15, 31] are mathematical schemes on account of asymmetric cryptography, mostly based on RSA algorithm. Given a message, the signer uses its private key to produce a signature. For authentication, the receiver verifies the reliability of the message by comparing the decoded value of signature and the hash value produced by the known public key.

Approaches based on MR-tree [23, 32] compute hash digests of all entries in a tree node firstly. Digests are computed in a bottom-up manner, and the single digest at the root is issued and signed by the DO. Then a client can verify through comparing the reconstructed root digest against the one that was signed by the DO. Although this method could effectively verify the query result, if any updates for the DO occur, all digests on the path from an affected leaf node to the root have to be recomputed. Other drawbacks are the high communication and processing overhead due to more I/O accesses.

For approaches based on Voronoi diagram [15], the DO shares the signed POI information with the LSP. When a user requests a query, the LSP processes the query based on an R-tree and returns the actual answer and additional Voronoi neighbors. From the signature and locations of the Voronoi neighbors, the user authenticates the reliability and correctness of the query answer. This method causes high storage overhead, as it stores all Voronoi neighbors for each POI. It also incurs a high communication overhead due to sending the same data information multiple times for authenticating different POIs. In order to address these limitations, a temporary buffer is introduced in [12]. Moreover, the DO only sends the count of Voronoi neighbors to the LSP. Our work follows this design, but we adapt the framework using R-tree proposed in [12] to GNN queries, and process spatial queries based on Voronoi diagram to make it more simple and efficient.

# 3 Problem statement

## 3.1 Preliminaries

We first present the definition of GNN query, relevant knowledge of AV-net protocol and Voronoi diagram.

**GNN query** Group nearest neighbor (GNN) query enables a group users to meet at a suitable place, such as restaurant or coffee shop [10]. Given a group of users $u_1, u_2, \cdots, u_n$ located at points $l_1, l_2, \cdots, l_n$, and $p_1, p_2, \cdots, p_m$ are POIs of the static point dataset. The formal definition is given below.

**Definition 1** (GNN Query) Let $U$ be a set of group users and $P$ be a set of POIs located in a 2-dimensional spatial space. A GNN query is that users jointly initiate a query to an LSP, and seek a POI with a smallest aggregate distance to all group members. Let $f$ be an aggregate function, such that the answered POI, $p$, meets the condition: for any $p' \in P - p$, $f(U, p) < f(U, p')$.

In general, the aggregate function of GNN query $f$ can be of different kinds. If it returns the POI which minimizes the sum of distances to the users, we call it *sum* GNN query. Similarly, it is a *maximum* GNN query when $f$ returns the POI which minimizes the maximum distance to the users. In this work, we mainly discuss the *sum* GNN query.

**AV-net protocol** In our approach, the idea of using *encrypted centroid* is based on an anonymous veto network (AV-net). Here, we briefly introduce the AV-net protocol, and more details can be found in [9].

Assume there are $n$ participating users, and they all agree on this protocol. Each user $u_i$ selects a random value as its secret: $a_i \in_R \mathbb{Z}_q$. In general, the AV-net protocol consists of two rounds. In the first round, each participating user $u_i$ publishes $g^{a_i}$ and a zero knowledge proof for $a_i$ to other users. When this round finishes, each user $u_i$ computes a masked value as:

$$g^{b_i} = \frac{\prod_{j=1}^{i-1} g^{a_j}}{\prod_{j=i+1}^{n} g^{a_j}}, 1 < i < n. \tag{1}$$

If $i = 1$, we let the equation $\prod_{j=1}^{i-1} g^{a_j} = 1$, and the equation $\prod_{j=i+1}^{n} g^{a_j} = 1$ when $i = n$, to make it meaningful.

In the second round, each user publishes a value $g^{c_i b_i}$ or $g^{a_i b_i}$, depending on whether the user $u_i$ vetoes or not. Moreover, $c_i$ is a random value satisfying $c_i \in_R \mathbb{Z}_q$ and a knowledge proof would also be given for $c_i$. Therefore, if no one votes, we have $\prod_i g^{a_i b_i} = 1$, because $\sum_i a_i b_i = 0$; if some users vote, we have $\prod_i g^{c_i b_i} \neq 1$. There are some properties for this protocol as shown bellow, which have been proved in [9].

- Soundness: For $a_i$ and $b_i$ defined in an AV-net, it holds that $\sum_i a_i b_i = 0$.
- Privacy: In an AV-net, $b_i$ is a secret random value to attackers in partial collusion against any other participating user.

**Voronoi diagram** Given a set of spatial data objects $P = \{p_1, p_2, \cdots, p_m\}$ in a space area, the Voronoi diagram of $P$ partitions the space area into $m$ disjoint regions. The most important property of Voronoi diagram is that each data object $p_i$ belongs to only one region and the principle of division is that all points in the region are closer to the corresponding $p_i$ than to any other point. Thus, each data object $p_i$ generates a region called the Voronoi cell, denoted as $VC(p_i)$. Therefore, the Voronoi diagram of $P$ is the union of all Voronoi cells. In addition, if two Voronoi cells have a common edge, the two data objects are regarded as Voronoi neighbors.

Here are some notable properties about Voronoi diagram [15, 20].

- Property 1: For a set of distinct objects $P = \{p_1, p_2, \cdots, p_m\}$, the Voronoi diagram of $P$ is unique.
- Property 2: The average number of Voronoi edges per Voronoi cell does not exceed six (i.e., the average number of Voronoi neighbors per object does not exceed six).

### 3.2 Assumptions and problem definition

We assume a 2-dimensional spatial database and Euclidean distance is used. Users' positioning devices can establish secure connection channels to LSP, so channel monitor attacks are not considered. The problem addressed in this paper can be described as: given a GNN query, we should find a correct and authenticated POI with a minimum aggregate distance of all group members, without exposing the exact locations of users to potential attackers. Besides, the locations of centroid point and meeting place are only visible to the GNN query participants.

### 3.3 System architecture

Figure 1 shows our system architecture. The system is composed of three parts: mobile users, LSP side and data owner.

First, we randomly select a coordinator inside the query group, who only knows the users' identities, the masked locations or cloaking regions, and the requested query type. Users cooperatively generate a centroid point, denoted as $q$, to represent all group members. Then, the coordinator sends the requested query type and the centroid point to the LSP, then receives and broadcasts the query result to group users.

On the other hand, a DO stores POIs in the form of Voronoi diagram, and signs a digital signature for each POI. Then, the DO shares these POIs to the LSP. After receiving a GNN query, the LSP processes the GNN query based on Voronoi diagram. After indexing the nearest POI of the centroid point, the LSP returns the answered POI, denoted as $p$, along

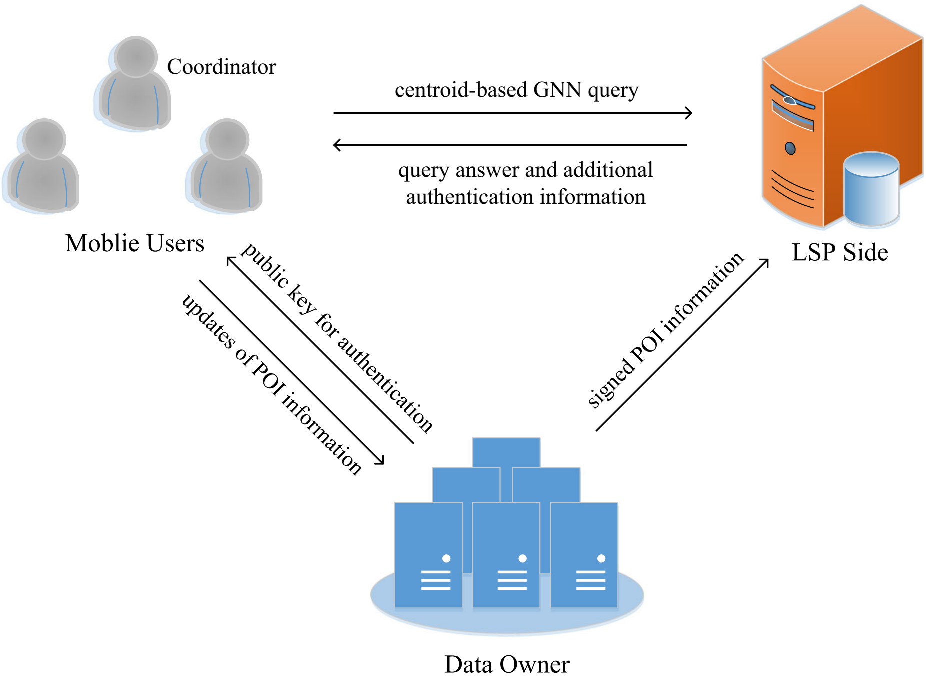

Coordinator
Moblie Users
centroid-based GNN query
query answer and additional authentication information
LSP Side
public key for authentication
updates of POI information
signed POI information
Data Owner

**Figure 1** System architecture: location privacy preservation & query result authentication

with additional POIs for authentication. Meanwhile, the LSP also stores such information in a temporary buffer, to avoid retrieving a same POI multiple times from the POI database.

Final step is authentication. Firstly, users would compute an authenticated known region, denoted as $AKR$. The authentication processing is divided into four parts: verifying digital signatures of POIs to ensure the reliability; verifying the *count* property of $p$ to ensure the completeness; ensuring $p$ is the correct answer with respect to the centroid point; demonstrating the answered POI $p$ is the correct answer to all group members. Otherwise, users need to put forward the query request to the LSP again, until returning a correct query result satisfying all authentication conditions.

## 4 Our approach

As can be seen in the system architecture, processing privacy preserving GNN queries can be divided into three major steps: (i) sending centroid-based query; (ii) processing query and returning answer along with additional POIs; and (iii) authenticating query result. In this section, we discuss them in details.

### 4.1 Sending centroid-based query

In the first step, we examine two different methods of centroid point generation.

**Cloaked centroid** In this method, each user cloaks location meeting pre-defined privacy requirements. As Figure 2 shows, a user $u_i$ will define a minimum cloaking region $R_{i,min}$

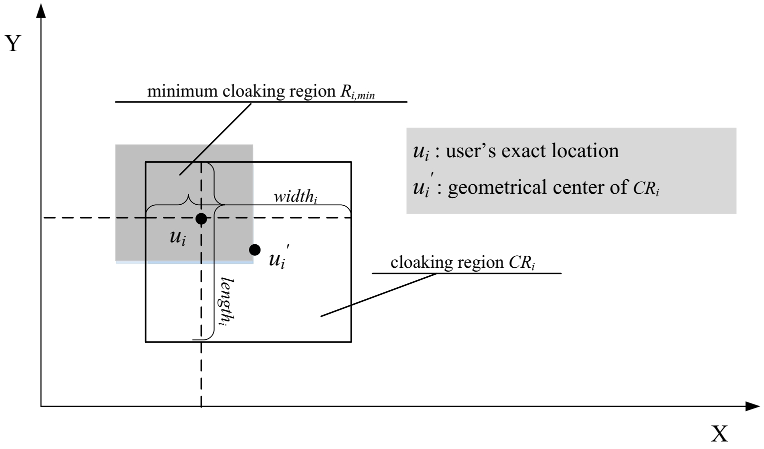

Y
X
minimum cloaking region $R_{i,min}$
$u_i$ : user's exact location
$u_i'$ : geometrical center of $CR_i$
$width_i$
$length_i$
$u_i$
$u_i'$
cloaking region $CR_i$

**Figure 2** Cloaking region meeting user requirements

first, represented by the gray rectangle in the figure, along with two fixed line segments ($width_i$ parallel to X-axis and $length_i$ parallel to Y-axis). Each user blurs actual location $l(u_i) = (x_i, y_i)$ into a cloaking rectangle $CR_i$ generated by the two line segments, where $u_i$ passes the two lines at any arbitrary point, and thus all points within $CR_i$ can be equally regarded as the user's location. Note that $CR_i$ should satisfy that $R_{i,min} \leq width_i \times length_i$ (i.e, the area of $CR_i$ should not be smaller than that of $R_{i,min}$). In addition, each $CR_i$ is represented by the coordinates of the up left and the bottom right points, denoted as $CR_i = \{(x_{i,u}, y_{i,u}), (x_{i,b}, y_{i,b})\}$, so as the geometric center of $CR_i$ is denoted by $u_i'$, which can be calculated by $l(u_i') = \left(\frac{(x_{i,u}+x_{i,b})}{2}, \frac{(y_{i,u}+y_{i,b})}{2}\right)$.

Then, each user sends $CR_i$ to the bulletin board through the secure connection channel, and users cooperatively generate a minimum region $MR$, which is determined by all users's cloaking regions, so that the centroid (rather than geometrical center) of $MR$ is regarded as the location of centroid, denoted by $q$. The representation of $MR$ is also given by the coordinates of the up left and the bottom right points, denoted as $MR = \{(x_{c,u}, y_{c,u}), (x_{c,b}, y_{c,b})\}$, computed as follows:

$$x_{c,u} = \frac{\sum_{i=1}^{n} x_{i,u}}{n}, x_{c,b} = \frac{\sum_{i=1}^{n} x_{i,b}}{n}. \tag{2}$$

Similar computation can be made for $y_{c,u}$ and $y_{c,b}$. Therefore, the centroid location of $q$ is calculated by $l(q) = \left(\frac{(x_{c,u}+x_{c,b})}{2}, \frac{(y_{c,u}+y_{c,b})}{2}\right)$.

**Encrypted centroid** This method also consists of two phases. In the first phase, each user masks exact location through utilizing the AV-net protocol, and gains a corresponding masked location. In the second phase, group users collaboratively compute the encrypted summations of all members' coordinates based on these masked locations through Paillier encryption [21].

According to previous definition of AV-net protocol, we compute mask value $g^{a_i b_i}$ for a user's location $l(u_i) = (x_i, y_i)$, and publish its masked location denoted as $l(u_i') = (g_s^{x_i} g^{a_i b_i}, g_s^{y_i} g^{a_i b_i})$.

Then, each user sends masked location $l(u_i')$ to the bulletin board, and users check the *soundness* of all AV-net masks in order to ensure the masked values have not been tampered.

If $\sum_i a_i b_i \neq 0$, we restart the encrypted centroid protocol in order to against active adversaries. After completing this simple validation, group users cooperatively generate the centroid point through multiplying all masked values, and use the Paillier encryption to encrypt the summations of the X and Y coordinates. The summations of all members' coordinates can be computed as follows:

$$\prod_{all\ x_i} g_s^{x_i} g^{a_i b_i} = g_s^{\sum x_i} modN^2, \prod_{all\ y_i} g_s^{x_i} g^{a_i b_i} = g_s^{\sum y_i} modN^2. \tag{3}$$

Then, the coordinator sends a single query containing the encrypted summation coordinates and the number of group members $n$ to the LBS. Upon receiving the query, the LSP decrypts it using the private key as follows:

$$Dec(w) = \rho = \frac{L(w^{\lambda} modN^2)}{L(g^s modN^2)} modN, \tag{4}$$

where $w$ is the cipher text and $\rho$ is the corresponding plain text. Then, the LBS divides the result by $n$ to get the coordinates of centroid $q$, and processes the requested GNN query based on it.

## 4.2 Processing query and returning answer

In the second step, we describe how LSPs process GNN queries and which query results would be returned to users.

**Generating digital signatures** Consider a DO has collected a large number of POIs for a certain space area. Each $p_i$ is in the form of $p_i = p_i.location, p_i.side, p_i.count, p_i.d)$ where the *location* property is a geographical coordinate represented by latitude and longitude, and the *side* property represents its descriptive information (e.g., name or type). Moreover, the DO computes the Voronoi neighbors of each $p_i$ with the count of Voronoi neighbors as another property $p_i.count$. In addition, the DO computes the distance between $p_i$ and the farthest Voronoi neighbor from itself, denoted as $p_i.d$. The value of $d$ measures the distance between the considered POI and its farthest Voronoi neighbor.

Then, the DO signs the digital signature of $p_i$. It uses its private key along with the information of $p_i$ to generate a digital signature, computed as $\kappa = sign(private\_key, h(p_i.location, p_i.side, p_i.count, p_i.d))$, where $h$ is a one-way, collision-resistance hash function and $sign$ is a function to generate the digital signature $\kappa$. $p_i$ has a form of $p_i = (p_i.location, p_i.side, p_i.count, p_i.d, p_i.\kappa)$.

**Processing GNN query** In our work, GNN query processing is conducted based on Voronoi diagram. To efficiently retrieve the nearest POI of centroid point $q$, a Voronoi diagram divides the space area into disjoint polygons, called Voronoi cells (VCs). Each $VC_i$ includes the nearest POI $p_i$ for any points inside. We can easily find the Voronoi cell containing $q$, and thus the generator of $VC_i$ is the nearest POI to $q$, denoted as $p$. $p$ is regarded as the returned query result for the GNN query. In addition, we use an R-tree to manage the Voronoi diagram on LSP side to improve retrieval efficiency.

**Returning answer** Besides returning $p$ to users, the LSP also needs to return some additional POIs for authentication. Firstly, the LSP computes a maximum search region, denoted as $MSR$. As Figure 3 shows, the centroid point $q$ sends a GNN query to the LSP, and $p$ is the answered POI with smallest distance from $q$. Let $f$ be the distance between $q$

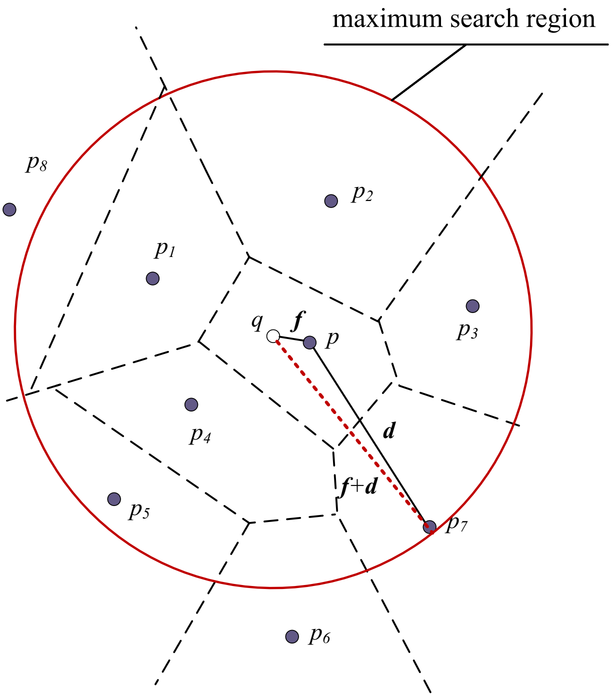


**Figure 3** An example of computing the maximum search region

and $p$, and $d$ is an attribute of $p$, which represents the distance between $p$ and its farthest Voronoi neighbor. Thus, the required maximum research region $MSR$ is a circle centered at $q$, along with the radius equals to the maximum search distance $|\mathbf{f}+\mathbf{d}|$. Finally, the LSP returns all other retrieved POIs located in $MSR$ and the entire information of $p$, including $location, side, count, d, signature.$

From Figure 3, it can be observed that some redundant POIs that are neither the answered POI nor the POIs for authentication could also been returned to users, such as $p_5$, as this retrieval is an incremental processing of GNN query until the correct answer is returned, and the LSP would store all returned information. Thus, if these redundant POIs are required for answering query or authenticating results later, there is no need to send these POIs again. Moreover, in order to ensure a single traversal that avoids retrieving the same POI multiple times from the database, the LSP always stores the retrieved information of POIs in a *temporary buffer* and keep tracks of $|\mathbf{f}+\mathbf{d}|$ as *covered search distance*. For example, if the answered POI $p$ has not passed one of the authentication conditions, we should retrieve the second nearest POI, which is $p_4$ in Figure 3. Although $p_4$ has already been sent to users, the additional POIs for authenticating $p_4$ may not have been sent. Thus, the LSP computes the required maximum research distance of $p_4$, denoted as $|\mathbf{f}'+\mathbf{d}'|$ based on the information of $p_4$, and computes a maximum search region of $p_4$, denoted as $MSR'$. Then, we check whether it is greater than the covered search distance of $p$. If yes, we should retrieve the information of POIs which are located out of $MSR$ but in $MSR'$, store them in the temporary buffer while also sending them to users, and finally update the values of *covered search distance* and *maximum search region*. Otherwise, there is no need to access the DO for retrieving more POIs. It is obviously that we can greatly reduce the communication and storage costs from the LSP side through the temporary buffer mechanism, and the termination condition of retrieval process is satisfied when the answered POI $p$ has passed all the validation steps.

### 4.3 Authenticating query result

In the third step, the authentication process is divided into four parts as follows.

**Authenticating reliability** There are two phases needed to ensure the reliability of $p$. On the one hand, group users decode the digital signature $p.\kappa$, and obtain the value of $h(p.location, p.side, p.count, p.d)$. On the other hand, users use the public key from the DO to compute the hash value of the information of POIs received from the LSP. It means to check whether it equals to the decoded value. If yes, the information of $p$ has not been tampered by adversaries. Conversely, we should drop $p$, and continue the incremental processing with the LSP.

**Authenticating completeness** The group users count the Voronoi neighbors of $p$ based on the returned additional POIs, and compare it with the *count* property of $p$. This operation ensures the LSP has returned all Voronoi neighbors of $p$, and we call it completeness of the result. Similarly, if the value computed by users equals to $p.count$, we continue authentication process. Otherwise, we should retrieve all Voronoi neighbors of $p$ from the POI database.

**Authenticating correctness through centroid point** First of all, the group users compute the distances for all the returned POIs from $q$, and then check whether $p$ is the nearest POI from $q$. This operation ensures $p$ is the correct answer based on the centroid point.

**Authenticating correctness through users' locations** The last but most important part is to check whether $p$ is the correct answer based on the group members' locations. Firstly, group users would compute an authenticated known region, denoted as $AKR$, which is a circle centered at the location of $q$, with radius equal to the distance between $q$ and the farthest authenticated POI from it, denoted as $p_a$. In addition, there might be more than one POI returned to users as the answered POI due to the incremental query request to the LSP, if the returned POI does not meet the authentication requirements. Therefore, a heap $H$ is maintained to store the incrementally authenticated POIs. The group users compute the sum of distances between their centroid points of $CR_i$ (for cloaked centroid) or masked locations (for encrypted centroid), namely $l(u_i')$, and any POI $p_i$ of all returned POIs, and then determine the POI with smallest distance sum, denoted as $p_0$. $p_0$ is the best meeting location in theory, although it is calculated on centroid points of $CR_i$ or masked locations, the relative distances are guaranteed, and $p_0$ must make the distance sum smallest. The circle, which centers at $q$ with the radius equals the maximum distance from $p_0$ to the group members, is theoretically the smallest area that needs authentication. We regard it as the necessary query region, denoted as $NQR$. If the authenticated known region $AKR$ contains $NQR$, it means $AKR$ formed by the returned $p$ has passed the authentication. That is to say, the answered POI $p$ is the correct answer to all group members. Otherwise, users push it to the heap $H$, and send the incremental query request to the LSP until the augmented authenticated known region $AKR$ contains $NQR$.

Algorithm 1 shows the authentication process in detail. The algorithm firstly constructs the authenticated known region $AKR$ from the GNN result, as lines 5-8 show. Then, users authenticate the answered POI $p$ in a sequential manner along with the additional POIs from the LSP. Lines 11–16 authenticate the reliability, lines 17–21 authenticate the completeness while lines 22–36 check whether $p$ is the nearest neighbor based on the centroid point $q$, and demonstrate whether $p$ is the correct answer based on all group users respectively. If $p$ has passed all the conditions, it is the authenticated correct answer.

**Algorithm 1** Authenticating query result

```
 1: $H \leftarrow$ the authenticated POIs;
 2: $R \leftarrow$ the set of returned additional POIs;
 3: $p \leftarrow$ the answered POI, $q \leftarrow$ the centroid point;
 4: $Q \leftarrow$ the cloaked or masked locations of users;
 5: while $H \neq \phi$ do
 6:     $p_{fa} \in H$ meeting $dist(q, p_{fa}) \geq dist(q, H)$;
 7:     $AKR \leftarrow circle\_construction(q, dist(q, p_{fa}))$;
 8: end while
 9: $flag \leftarrow true$;
10: while $p \neq null$ do
11:     $inf\_de \leftarrow Decode(p.\kappa)$;
12:     $inf\_en \leftarrow Encrypt(DO.publickey, p.location, p.side, p.count, p.d)$;
13:     if $inf\_de \neq inf\_en$ then
14:         $flag \leftarrow false$;
15:         break and retrieve POI;
16:     else if $inf\_de == inf\_en$ then
17:         $con \leftarrow Counting\_Voronoi\_neighbors(p, R)$;
18:         if $con \neq p.count$ then
19:             $flag \leftarrow false$;
20:             break and retrieve POI;
21:         else if $count == p.count$ then
22:             if $dist(q, p) \leq dist(q, R)$ then
23:                 while $R \neq \phi$ do
24:                     $p_0 \in R$ meeting $Sum\_dis(p_0, Q) \leq Sum\_dis(R, Q)$;
25:                     $NQR \leftarrow circle\_construction(q, dist(q, p_0))$;
26:                     if $NQR \subseteq AKR$ then
27:                         $p$ is the authenticated correct answer;
28:                         $H \leftarrow p$;
29:                     else if $NQR \nsubseteq AKR$ then
30:                         $flag \leftarrow false$;
31:                         break and retrieve POI;
32:                     end if
33:                 end while
34:             else if $dist(q, p) > dist(q, p')$ for $p' \in R$ then
35:                 $flag \leftarrow false$;
36:                 break and retrieve POI;
37:             end if
38:         end if
39:     end if
40: end while
```

## 5 Performance evaluation

In this section, we investigate security performance of two centroid-based preserving techniques mentioned above. We also compare the resource overhead of our approach with three competitive proposals.

### 5.1 Privacy analysis

We first investigate two different centroid point generation methods methods' behaviors under *collusion attack* and *knowledge attack*. In addition, the method of encrypted centroid allows *distorting attack* from active adversaries. In a collusion attack, some active adversaries may collude to discover the location(s) of some honest user(s) inside the group. A knowledge attack may take place when an adversary applies prior knowledge to infer users' true locations. A distorting attack means to broadcast fake values instead the AV-net masks in the method of encrypted centroid.

**Privacy analysis for cloked centroid** In this method, all points in the cloaking region of each user are equally likely to be the user's exact location, so an attacker cannot accurately infer the user's location. As static parameters, such as $width_i$, $length_i$ and $R_{i,min}$, are given by users themselves, a user is allowed to change settings in order to meet user-defined privacy protection requirements.

Assuming attackers have some prior knowledge about users' identities and users' approximate location areas, denoted as $ALR$. Although the attackers might get a user's cloaking region $CR_i$, if the area of $CR_i$ is smaller than prior knowledge $ALR_i$, it means they can get more information about the location of $u_i$, but not its exact geographical location. Moreover, if $CR_i$ is larger, the attackers gain no useful information. Thus, our cloaked centroid method can perform an effective resistance to *knowledge attack*.

Considering *collusion attack*, when the attackers collude with some query participants, since all broadcast information are users' cloaked rectangles, they get no advantage as well. However, if the number of group members is very small, it is easy for an attacker to speculate on the specific locations of the users, especially when all participants collude against only one user. To some extent, this is the limitation of geometrical cloaking.

**Privacy analysis for encrypted centroid** In this method, each user utilizes the AV-net protocol to gain a masked location and the summation of all members' coordinates is computed by the Paillier encryption. Then, the LSP decrypts the summation and divides the result by $n$ to get the coordinates of centroid $q$, so the location of $q$ does not reveal any useful information. Due to the cryptography of the summation and the randomness of the AV-net masks, users' location privacy from the *knowledge attack* can be totally preserved, which has been proved in [2].

On the other hand, assuming the attackers are trying to reveal the exact location, they need to remove the AV-net masked value $g^{a_i b_i}$ of $u_i$ firstly. However, $a_i$ is random and only generated and visible to $u_i$ herself, so as $b_i$ which is computed by the all group users. Even if an attacker and other group members are in collusion, the value of $a_i$ and $b_i$ cannot be obtained. Thus, the location privacy of group users is guaranteed against the *collusion attack*.

As mentioned above, the *distorting attack* means to broadcast fake values for the AV-net masks in our encrypted centroid method. However, due to the soundness property of AV-net protocol, the values of $a_i$ and $b_i$ need to meet the requirement $\sum_i a_i b_i = 0$. If attackers broadcast a fake value or try to modify the AV-net masks, this equation will not be established. Therefore, the malicious attacker would be detected before computing the encrypted centroid point, and the encrypted centroid protocol restarts. Therefore, the *distorting attack* is also resisted successfully.

**Table 1** POI's information storage cost of Hu et al. [15]

| POI property | Bytes |
|---|---|
| Location | 16 |
| Tail | 32 |
| Neighbors | 6*16 |
| Signature | 128 |
| Overhead per POI | 272 |

### 5.2 Storage comparison

To demonstrate how our approach reduces the resource overhead, we compare our approach with [15] while measuring the storage in bytes. Since each POI is stored as $p = (p.location, p.side, p.count, p.d, p.\kappa)$, it takes 182 bytes to store a single POI. Specifically, the $p.location$ attribute costs 16 bytes. For $p.d$ and $p.count$, it takes 4 and 2 bytes, respectively. For the signature attribute $p.\kappa$, it costs 128 bytes. Meanwhile, we propose to use 32 bytes to store the information of $p.side$. Therefore, the excepted overhead per POI is $16 + 32 + 4 + 2 + 128 = 182$ bytes (one 2-byte short integer, one 4-byte long integer, 16 bytes/point location, 32 bytes/side information, and 128 bytes/signature), which is a small storage overhead compared with the storage of POI information in [15] where each POI should be stored along with its Voronoi neighbors (at most six Voronoi neighbors per object), and thus the DO needs addition $6 * 16$ bytes to store Voronoi neighbors per POI object. More details are shown in Tables 1 and 2. The approach of [15] takes 49% more storage than our approach for storing each indivisual POI.

### 5.3 Experimental validation

To study how our approach reduces the *answer set size*, *communication cost* and *computation overhead*, the methods of *cloaked centroid* and *encrypted centroid* are both implemented, and we simulate our experiment on a system with Intel Xeon E5-2620 processor and 16GB of memory and compare their performances with two most related works: PGNN [10] and CCP [3]. We use Sequoia dataset[1] which contains 62,556 points of interest in California, and test the group size as 128, 265, 512 and 1024 with randomly generated users' locations. The detailed experimental results are shown as follows.

**Study on answer set size** In our method, the LSP processes a single query and returns a strong result (only the answered POI $p$ and some addition POIs for authentication). Since $MSR$ usually is not very large due to the Voronoi diagram, the number of additional POIs returned is also small. As Figure 4 shows, the answer set sizes of our cloked-centroid and encrypted-centroid methods can be consider as $\mathcal{O}(1)$ (i.e., do not change with group size). This is similar with the CCP method, because CCP processes query only in a small region, so the size of answer set is small. However, the PGNN method has to deal with all users' cloaking regions, so the size of answer set would be much larger, which shows a poor performance in terms of this aspect.

**Study on communication cost** Suppose there are $n$ group users, in our cloked-centroid method, users send $n$ messages to the bulletin board in total, and thus the communication

[1] http://chorochronos.datastories.org/?q=node/58

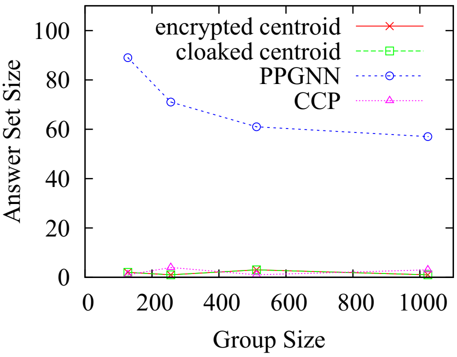
**Table 2** POI's information storage cost of our approach

| POI property | Bytes |
|---|---|
| Location | 16 |
| Neighbors | 32 |
| d | 4 |
| Count | 2 |
| Signature | 128 |
| Overhead per POI | 182 |

overhead is $\mathcal{O}(n)$. Besides, in our encrypted-centroid method, there are $2n$ messages to be sent in both two rounds, so the communication overhead is also $\mathcal{O}(n)$. After generating the centroid point $q$ (or $q_e$), the coordinator sends the requested query type and centroid point to the LSP, so the sending process's communication only costs $\mathcal{O}(1)$. This is the same as returning the answer result and even the additional POIs for authentication. Moreover, this communication cost is very small that can be ignored. As Figure 5 shows, both two methods are similar to the CCP method. CCP generates a cloaking region to send the query rather than a centroid point. However, the PGNN method tries to blur each user's location to a cloaking region, and all users send cloaking regions to the LSP, so the communication overhead of sending queries requires $\mathcal{O}(n)$. Besides, users send all their cloaking regions instead of simple geographical coordinates, whose communication overhead cannot be ignored. Moreover, for PGNN returned answer results are a set of POIs, so its communication cost is much larger. With the group size increasing, the communication cost of PGNN also increases more than ours and CCP.

**Study on computation overhead** As signing digital signatures is query independent and can be processed offline, we assume this computation cost could be ignored. Suppose there are $n$ users in the group. As for the cloaked centroid method, each user $u_i$ generates its cloaking region $CR_i$ firstly, so it takes $2n + 2$ addition operations and 4 division operations to get the centroid location of $q$ according to the computational process mentioned above. Obviously, this is a very small computation cost. As for the complicated encrypted centroid method, we measure its computation cost in exponentiation operations. In the phase of masking users' locations, it takes only one exponentiation to compute $l(u_i')$ while ignoring the multiplication operations; in the second phase of generating the masked centroid point $q_e$, it also requires one exponentiation operation. In general, our approach takes two

**Figure 4** Comparison of answer set size



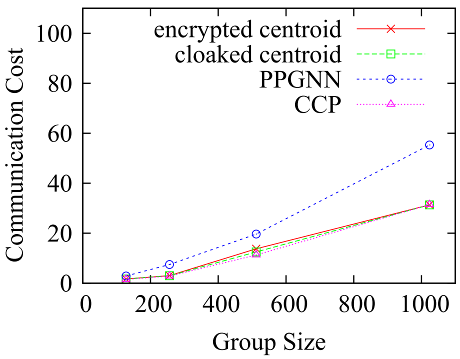


**Figure 5** Comparison of communication cost

exponentiation operations per user. On the LSP side, for the masked centroid point $q_e$, it is necessary to decode it first, as (4) shows. The computation cost requires two exponentiation operations. In the GNN query processing, finding nearest POI based on the centroid point is facilitated by a form of R-tree from the data owner, and its computation cost is represented as $\mathcal{O}(logN)$. Moreover, there is no need to filter out the correct answer, and the cost of authentication is small. As Figure 6 shows, the computation cost of CCP is also similar, due to the similar privacy preserving process. However, the PGNN method sends all users' cloaking regions to assess GNN queries, so the query processing of LSP could cause a huge burden due to a set of group regions. Moreover, the returned results are a set of candidate POIs, rather than only a strong result. Thus, the computation cost of PGNN is large. There is also a need to filter the answer set, which would lead to a further increase of computation cost.

**Effect of query authentication** In our approach, not only the answered POI $p$ but also $\eta$ additional POIs are returned for authentication. In terms of the reliability, it takes two exponentiation operations for the decryption of $p.signature$ generated by RSA algorithm and the complexity of $\mathcal{O}(1)$ to check whether the information is reliable. For the second step, the number of Voronoi neighbors of $p$ does not exceeds six, so the cost of counting process can be ignored. Besides, it also takes one operation to check. For the third step, it takes $\mathcal{O}(\eta)$ to check whether $p$ is closest to $q$. For the last step, it takes the complexity of $\mathcal{O}(n\eta)$ to compute the sum distance to all users' masked locations for all returned POIs, in order to check the correctness through users' locations. Therefore, our approach guarantees

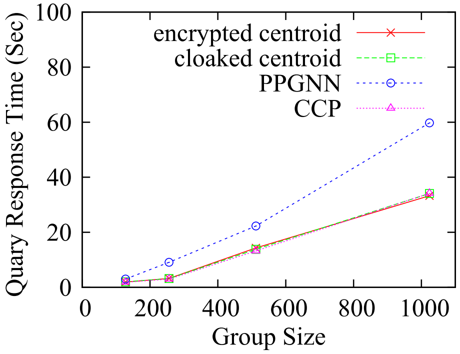


**Figure 6** Comparison of query response time

the correctness of the results, and the verification cost is small. It is worth noting that both PGNN and CCP do not consider the authentication of query results at all.

## 6 Conclusion

In this paper, we propose an effective system for processing GNN queries, which considers both location privacy preservation and query result authentication. Two centroid-based techniques are used to generate a centroid point to initiate GNN query, and an authentication algorithm based on Voronoi diagram is introduced. The analysis of security demonstrates how our approach is resistant to various attacks, and comparison of efficiency demonstrates our approach is better and more economical in terms of resource overhead.

**Acknowledgements** This work is supported by the National Natural Science Foundation of China (grants No. 61672133, No. 61632007 and No. 61602087), and the Fundamental Research Funds for the Central Universities (grants No. ZYGX2015J058 and No. ZYGX2014Z007).

## References

1. Andrés, M.E., Bordenabe, N.E., Chatzikokolakis, K., Palamidessi, C.: Geo-Indistinguishability: differential privacy for location-based systems. In: SIGSAC, pp. 901–914 (2013)
2. Ashouri-Talouki, M., Baraani-Dastjerdi, A., Selçuk, A.A.: GLP: a cryptographic approach for group location privacy. Comput. Commun. **35**(12), 1527–1533 (2012)
3. Ashouri-Talouki, M., Baraani-Dastjerdi, A., Selçuk, A.A.: The cloaked-centroid protocol: location privacy protection for a group of users of location-based services. Knowl. Inf. Syst. **45**(3), 589–615 (2015)
4. Chen, Q., Hu, H., Xu, J.: Authenticating top-k queries in location-based services with confidentiality. PVLDB **7**(1), 49–60 (2013)
5. Fu, Z., Shu, J., Wang, J., Liu, Y., Lee, S.: Privacy-preserving smart similarity search based on simhash over encrypted data in cloud computing. J. Internet Technol. **16**(3), 453–460 (2015)
6. Gedik, B., Liu, L.: Location privacy in mobile systems: a personalized anonymization model. In: ICDCS, pp. 620–629 (2005)
7. Ghinita, G., Kalnis, P., Skiadopoulos, S.: Mobihide: a mobilea peer-to-peer system for anonymous location-based queries. In: SSTD, pp. 221–238 (2007)
8. Ghinita, G., Kalnis, P., Khoshgozaran, A., Shahabi, C., Tan, K.: Private queries in location based services: anonymizers are not necessary. In: SIGMOD, pp. 121–132 (2008)
9. Hao, F., Zielinski, P.: The power of anonymous veto in public discussion. Trans. Computational Science **4**, 41–52 (2009)
10. Hashem, T., Kulik, L., Zhang, R.: Privacy Preserving Group Nearest Neighbor Queries. In: EDBT, pp. 489–500 (2010)
11. Hashem, T., Ali, M.E., Kulik, L., Tanin, E., Quattrone, A.: Protecting privacy for group nearest neighbor queries with crowdsourced data and computing. In: Ubicomp, pp. 559–562 (2013)
12. Hashem, T., Datta, S., Islam, T.U., Ali, M.E., Kulik, L., Tanin, E.: A unified framework for authenticating privacy preserving location based services. In: Georich@SIGMOD 2015, pp. 13–18 (2015)
13. Hu, H., Xu, J.: Non-exposure location anonymity. In: ICDE, pp. 1120–1131 (2009)
14. Hu, H., Xu, J., Chen, Q., Yang, Z.: Authenticating location-based services without compromising location privacy. In: SIGMOD, pp. 301–312 (2012)
15. Hu, L., Ku, W., Bakiras, S., Shahabi, C.: Spatial query integrity with voronoi neighbors. IEEE Trans. Knowl. Data Eng. **25**(4), 863–876 (2013)
16. Huang, Y., Vishwanathan, R.: Privacy preserving group nearest neighbour queries in location-based services using cryptographic techniques. In: GLOBECOM, pp. 1–5 (2010)
17. Khan, A.K.M.M.R., Hashem, T., Tanin, E., Kulik, L.: Location Oblivious Privacy Protection for Group Nearest Neighbor Queries. In: GIScience, pp. 301–317 (2014)
18. Khoshgozaran, A., Shahabi, C.: Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy. In: SSTD, pp. 239–257 (2007)

19. Kido, H., Yanagisawa, Y., Satoh, T.: An anonymous communication technique using dummies for location-based services. In: ICPS, pp. 88–97 (2005)
20. Okabe, A., Satoh, T., Furuta, T., Suzuki, A., Okano, K.: Generalized network voronoi diagrams: concepts, computational methods, and applications. Int. J. Geogr. Inf. Sci. **22**(9), 965–994 (2008)
21. Paillier, P., Pointcheval, D.: Efficient public-key cryptosystems provably secure against active adversaries. In: ASIACRYPT, pp. 165–179 (1999)
22. Papadias, D., Shen, Q., Tao, Y., Mouratidis, K.: Group nearest neighbor queries. In: ICDE, pp. 301–312 (2004)
23. Papadopoulos, S., Yang, Y., Bakiras, S., Papadias, D.: Continuous spatial authentication. In: SSTD, pp. 62–79 (2009)
24. Sadeghi, A., Visconti, I., Wachsmann, C.: Anonymizer-enabled security and privacy for RFID. In: CANS, pp. 134–153 (2009)
25. Sheikh, R., Mishra, D.K., Kumar, B.: Secure multiparty computation: from millionaires problem to anonymizer. Information Security Journal: A Global Perspective **20**(1), 25–33 (2011)
26. Shen, J., Liu, D., Shen, J., Liu, Q., Sun, X.: A secure cloud-assisted urban data sharing framework for ubiquitous-cities. Pervasive Mob. Comput. (2017). https://doi.org/10.1016/j.pmcj.2017.03.013
27. Shen, J., Shen, J., Chen, X., Huang, X., Susilo, W.: An efficient public auditing protocol with novel dynamic structure for cloud data. IEEE Trans. Inf. Forensics Secur. **12**(10), 2402–2415 (2017)
28. Song, J., Shen, H.T., Wang, J., Huang, Z., Sebe, N., Wang, J.: A distance-computation-free search scheme for binary code databases. IEEE Trans Multimedia **18**(3), 484–495 (2016)
29. Sweeney, L.: k-anonymity: a model for protecting privacy. Int. J. Uncertainty Fuzziness Knowledge Based Syst. **10**(5), 557–570 (2002)
30. Um, J., Kim, Y., Lee, H., Jang, M., Chang, J.: k-nearest neighbor query processing algorithm for cloaking regions towards user privacy protection in location-based services. Journal of Systems Architecture - Embedded Systems Design **58**(9), 354–371 (2012)
31. Xu, S., Yang, W., Lau, F.C.M.: A visualization based approach for digital signature authentication. Comput. Graph. Forum **28**(3), 935–942 (2009)
32. Yang, Y., Papadopoulos, S., Papadias, D., Kollios, G.: Authenticated indexing for outsourced spatial databases. VLDB J. **18**(3), 631–648 (2009)
33. Yiu, M.L., Ghinita, G., Jensen, C.S., Kalnis, P.: Enabling search services on outsourced private spatial data. VLDB J. **19**(3), 363–384 (2010)
34. Zhu, X., Zhang, L., Huang, Z.: A sparse embedding and least variance encoding approach to hashing. IEEE Trans. Image Process. **23**(9), 3737–3750 (2014)