Hindawi
Security and Communication Networks
Volume 2019, Article ID 1914038, 11 pages
https://doi.org/10.1155/2019/1914038

Research Article

A Novel Method for Location Privacy Protection in LBS Applications

Dan Lu,[1] Qilong Han,[1] Kejia Zhang,[1] Haitao Zhang,[1] and Bisma Gull[2]

[1]*College of Computer Science and Technology, Harbin Engineering University, Harbin, 150001, China*
[2]*Department of Electronic and Communications, National Institute of Technology Srinagar, Jammu and Kashmir, 190006, India*

Correspondence should be addressed to Kejia Zhang; kejiazhang@hrbeu.edu.cn

Received 17 April 2019; Accepted 16 June 2019; Published 15 July 2019

Guest Editor: Houbing Song

Location-based services have become a mainstream in people's daily lives due to continuous innovations in the field of mobile networking and GPS technologies. Recently they have advanced into a hot topic to which the majority of researchers pay close attention about how to enjoy them while safeguarding the location privacy of mobile users. Existing works involve the injection of random noise that cannot pledge the quality of service. Herein this manuscript, we propose a novel location privacy protection model based on the loss of service quality. This model allows the user to express his/her requirement of service quality by specifying the maximum service quality loss L_{max}, which is the user's tolerance. L_{max} can be set to 0. Our comprehensive experimental evaluation using a real-world dataset demonstrates that our modus outdoes other state-of-the-art approaches.

1. Introduction

Location-based services (LBS) are swelling owing to the innovations in technology and the dominance of location-cognizant devices [1–3]. Such services take the user's location information in a query as an input, execute the query at the server, and then provide the user with the information of nearby points of interest (POI), such as gas stations, banks, and restaurants [4]. A wide range of LBS applications include location-aware search (Baidu Maps), E-commerce (Meituan, Dianping), location-based social recommendation (QQ, WeChat), and ordering application and crowdsourcing (Ali) [5].

During this process, users' current or future whereabouts and interests are disclosed to the LBS server through their queries. Access to all submitted information is deemed necessary to best serve users; the LBS server is entrusted with rich information [6–8]. However, many studies have revealed that service providers can be honest but curious, belligerently stockpiling information of profile users, identifying homes, working places, and social relationships, or inferring interests towards commercial purposes [9–12]. Therefore, the concern of LBS is to provide high quality service while the user's location is anonymous to the LBS server. It seems contradictory and challengeable [13].

The topic of LBS privacy has been widely studied. In 2003, Beresford proposed the concept of location privacy [14], which pioneered the research on location privacy protection. Since then, the research on discrete location privacy or trajectory privacy has been published successively. There are two types of privacy issues in LBS: location privacy and query privacy [15]. Location privacy includes users' previous, current, and future location and query privacy is the type of POI he/she is interested in. In addition, the importance of query privacy is greater when the request is sensitive (query for hospitals). In this paper, we propose a novel location privacy protection model for the former, which ensures high quality service while user location is protected.

These approaches regarding the location privacy in LBS are classified into three categories. The first one is to enlarge the user location into a region; the representative is k-anonymity: e.g., an obfuscated region is formed by k users [16]. The second one can be viewed as a dummy-based technique [17]. The dummy location is sent to the LBS server instead of the exact location. Obviously, the user utilizes another position to replace his/her location. The major limitation of such replacement is that the quality of service degrades significantly if the user chooses a higher level of privacy. The last one is to transform the original query into another problem such that the users' location cannot be inferred [15]. This kind of approaches usually employs cryptographic algorithms and spatial transformation techniques (e.g., Hilbert curve).

Geo-indistinguishability (GeoInd), a formal notion of location privacy introduced in [18], builds on the concept of differential privacy [19] to design user-centric location privacy protection mechanisms. GeoInd guarantees that obfuscated locations are statistically indistinguishable from other locations within a radius around the users' real location. However, [20] illustrates that GeoInd can be misleading in terms of both privacy and utility. Sometimes, GeoInd mechanisms possibly generate an obfuscated location vary far away from the user leading to dissatisfying service quality.

In this paper, a trusted third party (TTP) is added between the user and the server, for collecting users' real location and then sending to the LBS server with the disturbed one. The TTP perturbs the real location with a novel location privacy protection model based on loss of service quality which solves the challenge to ensure high quality service while protecting location privacy. To summarize, our contributions are as follows:

(1) We propose a function of service quality loss ($loss(P_n, P_r)$) based on a real query result, in which P_r is the real query result and P_n is the perturbation

(2) We propose a novel location privacy protection model based on loss of service quality. The TTP calculates a noisy area based on the maximum service quality loss (L_{max}) specified by the user and selects one point randomly to return to the server

(3) We propose a novel traversal method based on a Voronoi diagram, considering the geographic relationships of locations that efficiently reduce the complexity of computation

2. Related Works

Due to the paramount importance of location privacy in LBS services, it has been studied extensively, and several methods have been proposed. We review some of the typical briefly. K-anonymity based on trajectory generalization has been prevailing for its good balance of privacy protection and data availability. In [21], a (k, δ) anonymous algorithm was proposed for trajectory dataset publication. Based on trajectory generalization and k-anonymity, this algorithm generalizes every position in the trajectory to a circle with a radius of δ and makes sure that each circle has at least k points to satisfy k-anonymity, each of which is represented by a cylinder of these circles. Literature [22] proposed a technique by replacing the original data with a logical one.

Differential privacy was quickly applied to the privacy protection of data publishing [23] based on fake data technology to achieve privacy protection by adding noise to the real dataset [24]. In data distribution, differential privacy can achieve different levels of privacy protection and data

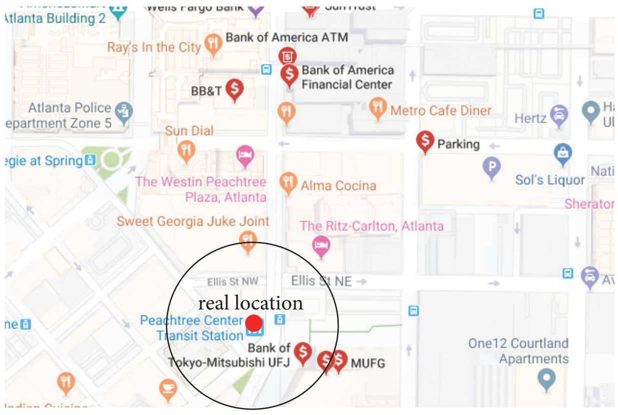

real location
Bank of America ATM
Bank of America Financial Center
BB&T
Bank of Tokyo-Mitsubishi UFJ
MUFG
Peachtree Center Transit Station
Ellis St NE
Ellis St NW
The Westin Peachtree Plaza, Atlanta
The Ritz-Carlton, Atlanta
Sweet Georgia Juke Joint
Alma Cocina
Sun Dial
Ray's In the City
Metro Cafe Diner
Parking
Sol's Liquor
Hertz
Atlanta Police Department Zone 5
One12 Courtland Apartments
Wells Fargo Bank
SunTrust
Atlanta Building 2

Figure 1: Actual result of banks within 500 m.a

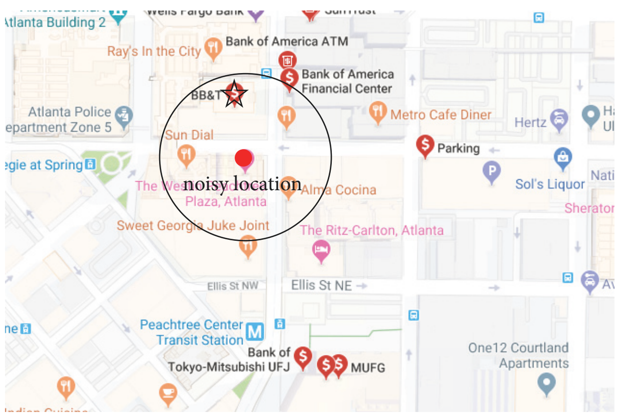

noisy location
Bank of America ATM
Bank of America Financial Center
BB&T
Bank of Tokyo-Mitsubishi UFJ
MUFG
Peachtree Center Transit Station
Ellis St NE
Ellis St NW
The Westin Peachtree Plaza, Atlanta
The Ritz-Carlton, Atlanta
Sweet Georgia Juke Joint
Alma Cocina
Sun Dial
Ray's In the City
Metro Cafe Diner
Parking
Sol's Liquor
Hertz
Atlanta Police Department Zone 5
One12 Courtland Apartments
Wells Fargo Bank
SunTrust
Atlanta Building 2

Figure 2: Query result with a noisy location.

publishing accuracy by adjusting the privacy parameter ϵ. In general, the larger the value of ϵ, the lower the level of privacy protection and the higher the accuracy of the published dataset. The first common mechanism for implementing differential privacy is the Laplace mechanism proposed in [25]. This mechanism mainly focuses on numeric queries, by adding random noise obeying the Laplace distribution to the results of real queries [26, 27]. For nonnumeric queries, [28] proposed an exponential mechanism, which is the second universal mechanism to achieve differential privacy.

Since its proposal in 2013, GeoInd has drawn a lot of attention from the research community. It has been widely used based on its core qualitative advantages, regardless of the adversary's background information. However, [21] illustrates that GeoInd is not that great. Sometimes, GeoInd possibly generates an obfuscated location very far away from the user leading to worthless data as shown in Figures 1 and 2.

To rectify this, we propose a novel location privacy protection model to replace the user's real location with an obfuscated one based on loss of service quality.

3. Preliminaries

In this section, the symbols and related definitions used in this paper are given. As mentioned earlier, the quality

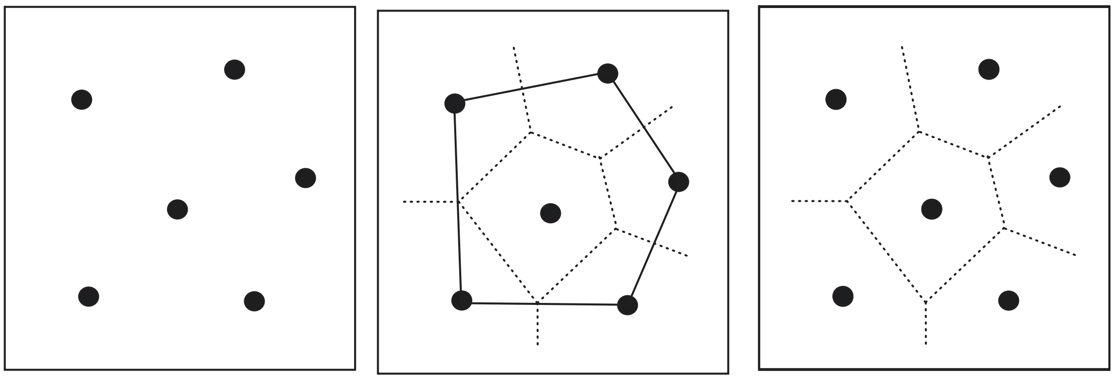


Figure 3: Generation of the Voronoi diagram.

of service declines dramatically after adding the Laplace noise, which means the obfuscated location is far away from the real one. To fix this, we propose a loss of service quality ($loss(P_n, P_r)$) based on the real query result as a novel evaluation index. The obfuscated location is generated randomly from the noisy area, which is calculated according to a specific $loss(P_n, P_r)$.

Real query result: the TTP receives the realistic location l and query radius r of a user, using l as the center within r, the set made up of points of interest (POI) sorted by the distance from l.

LBS query result: the LBS server takes the obfuscated location l' from the TTP, using l' as the center within same r, the set made up of points of interest (POI) sorted by the distance from l'. This article leverages the maximum service quality loss (L_{max}) to constrain the LBS query result.

Loss of service quality ($loss(P_n, P_r)$): regard the change of the real query result as $loss(P_n, P_r)$. According to the statistics about the clickthrough rates of search results released from AOL and IMN [29], ranking and attention were found to be expressed by a power function $y = \lambda a^{-x} (a > 1)$. Therefore, the weight of rank is set to $W(1) = \lambda a^{-1}$, $W(2) = \lambda a^{-2}, \ldots, W(k-1) = \lambda a^{1-k}$, $W(k) = \lambda a^{1-k}$, in which $w(i)$ denotes the weight of rank i, and the last two weights repeat. Given the real query result $P_r =< a, b, c, \ldots)$ and the obfuscated result $P_n =< \ldots, a, \ldots, b, \ldots)$, in which rank (a, P_r) denotes the index of POI a at P_r, $loss(P_n, P_r)$ is formally defined below: compare the ranking of each POI after added noise; regard the weight difference $w(\text{rank}(x, P_r)) - w(\text{rank}(x, P_n))$ as the loss of x if the ranking drops. We set $w(\text{rank}(x, P_n)) = 0$ if x is not present in the obfuscated result.

$$loss(P_n, P_r) = \sum_{x \in P_r} \Delta_w(x)$$

$$\Delta_w(x) = \begin{cases} w(\text{rank}(x, P_r)) - w(\text{rank}(x, P_n)) & \textit{if } \text{rank}(x, P_r) < \text{rank}(x, P_n) \\ 0 & \textit{else} \end{cases} \tag{1}$$

$$\textit{For } x \notin P_n,\ \text{rank}(P_n) = +\infty,\ w(\text{rank}(x, P_n)) = 0.$$

Maximal tolerance L_{max}: this is the maximal loss of service quality that a user may tolerate. The smaller L_{max} is, the more similar the obfuscated result and the real one are.

Euclidean distance: the Euclidean distance is the shortest distance between two points in space. Given two points in two dimensions (x_1, y_1) and (x_2, y_2), the Euclid distance of two points is defined: $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$.

The Voronoi diagram [30]: the Voronoi diagram, also known as the Thiessen polygon or Dirichlet diagram, generates a Delaunay triangulation at first and connects the center of the circumcircle of the adjacent triangle. The characteristic is that there is a generator with each V polygon in the graph, and the distance from the inner point of each V to the generator is shorter than other generators. Points on the boundary of two polygons are equidistant from the corresponding generator. The establishment method of the Voronoi diagram is shown in Figure 3.

4. Our Framework of Privacy

In this section we describe our system architecture and the novel method of location privacy protection. We use Baidu Maps API [31] as the trusted third party (TTP) which is added between the user and the server, for collecting the user's real location and then sending to the LBS server with the disturbed one. The TTP perturbs the real location with a novel location privacy protection model based on loss of

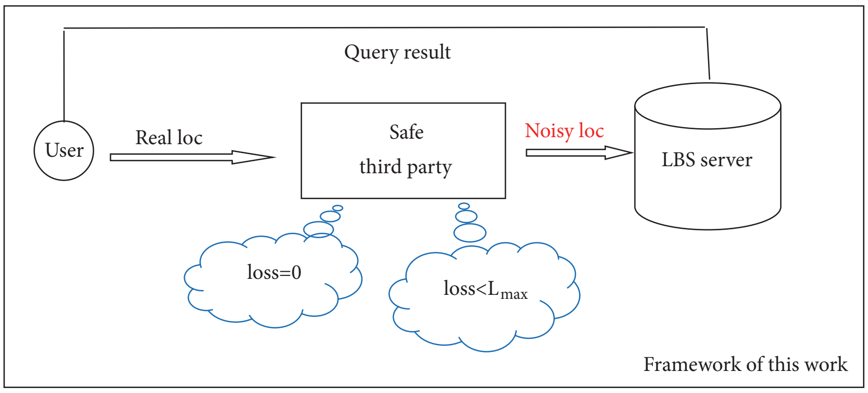


Figure 4: System architecture.

service quality which solves the challenge to ensure high quality service while protecting location privacy. The overall system architecture is shown in Figure 4. The model allows a user to express his/her requirement of service quality by specifying a maximum service quality loss L_{max}, in which the user would tolerate the loss of service quality ($loss(P_n, P_r) \leq L_{max}$). L_{max} can also be set to 0, which means immutable service quality. To guarantee the quality of query service, L_{max} is typically set to a small value.

It can be easier to calculate the distance between points in two dimensions; we convert from latitude and longitude to UTM coordinates [32], also flagged as l.

4.1. Nonlossy Service Quality ($loss(P_n, P_r) = 0$). As mentioned in the previous section, the loss of service quality ($loss(P_n, P_r)$) based on the real query result is a novel evaluation index to measure the difference between the real query result and the obfuscated one. There are two kinds of situations. The first is that the number of POI and ranking stay the same (the last two points of interest are interchangeable). In another story, the number of POI increases while the rankings of points in the real query result are interchangeable. For instance, given real query result *ABCDE*, the obfuscated result could be *ABCDE* or *ABCDE***FG** if $loss(P_n, P_r) = 0$. POI *F* and *G* in bold are generated in addition without impacting the loss of service quality. Therefore, to ensure nonlossy service quality, postprocessing would be required on the obfuscated result.

4.1.1. Generation of Obfuscated Region. Given a real location l of a user, the real query result can be obtained by calling Baidu Maps API. Calculate the obfuscated region according to the ranking of the real query result (proximal to distal from l) and the Voronoi diagram. The distance from each point within it to each query result satisfies the true ranking.

Algorithm 1 illustrates the details of the algorithm of obfuscated region generation. Given the real location l of a user, Step 2 obtains the real query result by calling Baidu Maps API. Step 4 executes the Delaunay triangulation algorithm backwards according to the ranking. We compute the overlapped region at each step to find the final obfuscated region κ. The step of the Delaunay triangulation algorithm is as follows:

```
Input: Real location $l$, radius $r$
Output: generated area $\kappa$
1. Initialize generated area $\kappa = \phi$
2. $P_r = BaiduAPI.Query(l)$
3. for each $x \in P_r$ do
4.     Generate Delaunay triangulation of $x$
5.     Calculate the Voronoi area of $x$ as $\kappa_x$
6.     $P_r.remove(x)$
7.     if $x = top1$ then
8.         $\kappa = \kappa_x$
9.     else
10.        $\kappa = \kappa \cap \kappa_x$
11. $\kappa = \kappa \cap$ circle($P_r.getLast(), r$)
12. return $\kappa$
```

Algorithm 1: Obfuscated region generation.

(1) Construct the Delaunay networks with the discrete points

(2) Calculate the center of the circumcircle of each triangle and take it down

(3) Look for three adjacent triangles whose border is in common with the current triangle

(4) If adjacent triangles are found, connect the circumcenter of each one to the circumcenter of the current triangle. If not, calculate the midperpendicular of the outermost border

4.1.2. Postprocessing of Obfuscated Region. We get the obfuscated region from the previous section satisfying $loss(P_n, P_r) = 0$, like polygon *AOPQR* in Figure 5. A closer inspection would reveal an extra POI *K* on the certain extension of the query if the obfuscated location were located on *A*. Besides, the distance from *K* to *A* is less than the distance from *D* to *A* ($d_2 < d_1$). In this case, *K* influences the ranking of *D* leading to $loss \neq 0$. So, to hedge against this, we need to do the postprocessing of our region.

To guarantee $loss(P_n, P_r) = 0$, the distance from *K* to the obfuscated location l' must be larger than the distance from the last-ranking POI *D* to l', which is ($dis(K, l') \geq dis(D, l')$).

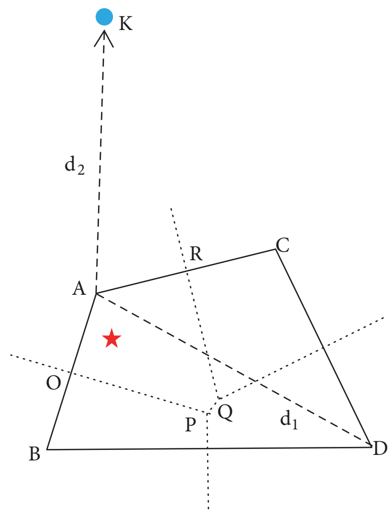


Figure 5: Obfuscated region from Section 4.1.1.

```
Input: Generated area $\kappa$, radius $r$, Real Rank $P_r$
Output: Final area $\kappa'$
1. Let V=vertex of area $\kappa$, $N = \phi$, $\kappa' = \kappa$
2. init A=$\phi$
3. for each $v \in V$ do
4.     Circle with radius $r$ as $c_v$(v=1 to $|V|$)
5. Make the tangent of $\begin{cases} (c_v, c_{v+1}) & v \leq |V| \\ (c_v, c_1) & v = |V| \end{cases}$
6. end for
7. A=All enclosed area
8.     $QueryResult(Q) = BaiduAPI(A)$
9.     if $\exists p \in Q \cap p \notin P_r$
10.        $N.add(p)$
11. for each $n \in N$ do
12.     $\kappa'$=Area surrounded by perpendicular bisector
          of $(n, P_r.getLast())$ and $\kappa'$
13. end for
14. return $\kappa$
```

Algorithm 2: Postprocessing step.

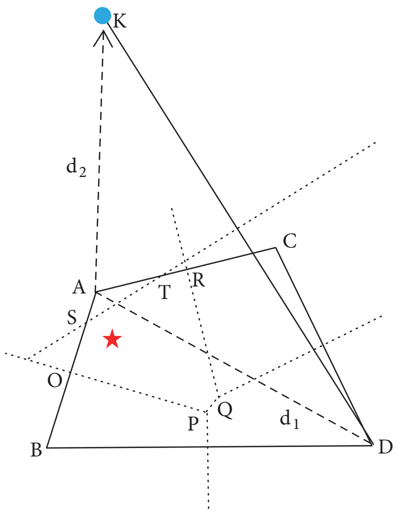


Figure 6: Ultimately obfuscated region.

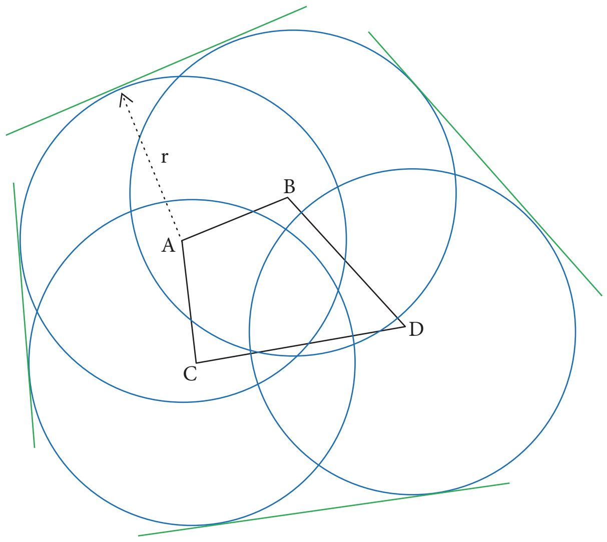


Figure 7: Query ranges.

The vertical bisector of segment KD crosses polygon $AOPQR$ at POI S and T. We get the ultimately obfuscated region just like Figure 6.

Algorithm 2 illustrates the details of the postprocessing algorithm. Given the obfuscated region κ calculated in the previous section, we initialize the set of vertices as V and the set of extra points $N = \phi$. In Steps 4 to 6, we compute the query ranges A with query radius r. The area enclosed by the red line in Figure 7 is the query ranges. Decide whether there come extra points after calling Baidu Maps API again within A and add them to N (Steps 7–9). In Steps 10 to 12, for each point n in the set N, draw the vertical bisector of segment $(n, P_r.getLast())$ crossing κ to form the new κ. That can ensure the distance from the last-ranking to the obfuscated location is less than that of any point in N to the obfuscated location.

4.2. Tolerable Quality of Services ($loss(P_n, P_r) \leq L_{max}$).

In this section, we consider the case that service quality might be lossy. Given the maximal tolerance L_{max}, the loss of service quality within it is acceptable. By our definition in Section 3, the loss of service quality is the loss of weight regarding the real query result. The top priority in this section is to find all possible rankings under satisfying constraints, expressed as $P_x \mid loss(P_x, P_r) \leq L_{max}$. The traditional enumeration is complex, so we propose two enhanced enumeration algorithms to reduce the time complexity effectively.

4.2.1. Enumeration with Pruning.

The first algorithm is enumeration with pruning. For a certain position i, calculate the upper bound sup L and the lower bound inf L of the queue

```
Input: real ranking $P_r$, Maximum tolerance $L_{max}$
Output: ranking Set $P$
1. init $P = \phi$, sup($loss$), inf ($loss$), $C = P_r \cup \chi$
2. for each $c \in C$ do
3.     calculate sup $L(c)$ and inf $L(c)$
4.     if (inf $L(c) < L_{max}$) then
5.         $P.append(c)$
6. for $i = 2$ to $|P_r|$ do
7. init $N = \phi$
8.     for each $p$ in $P$ do
9.         for $j = 1$ to $|P_r| + 1$ do
10.          if ($pcontains(C[j]) \cap C[j] \neq \chi$) then
11.              continue
12.          temp = p + C[j]
13.          calculate sup $L(temp)$ and inf $L(temp)$
14.          if (inf $L(temp) < L_{max}$) then
15.              $N.append(temp)$
16.     $P = N$
17. return $P$
```

Algorithm 3: Enumeration with pruning.

```
Input: real ranking $P_r$; Maximum tolerance $L_{max}$
Output: ranking Set $P$;
1. init queue $q$, set of Candidate $Candy$;
2. $list = P_r$; 3. for each $x \in P_r$ do
4.     Generate Delaunay triangulation of $x$
5. end for
6. function circulate($list$):
7.     for each $item \in list$ do
8.         q.add(item);
9.         if sup $L(q) \leqslant L_{max}$ then
10.            $P.add(q)$;
11.        else if inf $L(q) > L_{max}$ then
12.            candy.remove(x) rank ($x$) > rank ($item$);
13.        else
14.            q.add(item);
15.            calculate the candidate of $q$;
16.            circulate($Candy$);
17. return $P$;
```

Algorithm 4: Enumeration with the Voronoi diagram.

after POI x joined. The POI x is not allowed in position i ($i \leq |P_r|$) if inf $L > L_{max}$. Therefore, we can prune the branch of rank $(x) = i$. This is as in Algorithm 3.

Algorithm 3 illustrates the details of the enumeration algorithm with pruning. Given the real ranking P_r, we consider the possibility that each point may be the first. In Steps 6 to 9, we add POI into queue $temp$ in turn to calculate sup $L(temp)$ and inf $L(temp)$. Each point can appear only once and the extra points may occur several times (Steps 10-11). In Steps 13 to 15, we store the current queue to N if the lower bound inf L is less than maximum tolerance L_{max}. By analogy, all the rankings that meet the constraints are obtained. This method has no regard for the geospatial and will generate many rankings unable to form a region.

4.2.2. Enumeration with a Voronoi Diagram. The pruning algorithm also has a high time complexity, and it will generate many useless rankings which can not form a region. To solve this, an enumeration method with a Voronoi diagram is given in this paper. The ultimately obfuscated region satisfying L_{max} can be obtained by dividing the polygons continuously. This method operates on the Voronoi diagrams directly, which is intuitive and easier for getting the obfuscated region without postprocessing.

Algorithm 4 illustrates the details of the enumeration with the Voronoi diagram algorithm. Given the real ranking P_r, we generate the Voronoi diagram only once. Step 6 starts the recursive function; calculate the upper bound sup $L(q)$ and the lower bound inf $L(q)$ after each addition. If the condition sup $L(q) \leqslant L_{max}$ is met, add the current queue q into ranking set P. Moreover, if the condition inf $L(q) > L_{max}$ is met, we remove all the points in candidate set $Candy$ that ranked lower than q. Besides, in Steps 13 to 16, we divide the current region into multiple regions and start a new round of recursion. The candidate set creating algorithm is as in Algorithm 5.

Algorithm 5 illustrates the details of the candidate generation algorithm. Given the current queue q, the Delaunay triangulation dt, and the current candidate set $Candy$, the algorithm finds the neighbor POI of each point in queue q and adds them to $Candy$. Then, it sorts them according to the raw ranking. Using Figure 8 as an example, the real ranking is $\langle A, B, C, D\rangle$ and the maximum tolerance $L_{max} = 0.1$. Generate the Voronoi diagram at first, and calculate the likelihood of a given queue with each point on the top, just like $\langle A \cdots\rangle, \langle B \cdots\rangle, \langle C \cdots\rangle, \langle D \cdots\rangle$. The lower bounds of B,C,D inf $L(B \mid C \mid D)$ are all greater than 0.1; that is why the top one must be A. After that, we partition the V polygon $Aqwry$ into smaller polygons; the vertical bisector of segment BC crosses $Aqwry$ at points t and e. Calculate the upper bound and the lower bound of two polygons, which are sup $L(AB \mid AC)$ and inf $L(AB \mid AC)$. The lower bound of polygon $tyre$ is beyond L_{max}, while the upper bound of polygon $Aqwet$ is under L_{max}. Therefore, we regard the gray area $Aqwet$ as the ultimately obfuscated region.

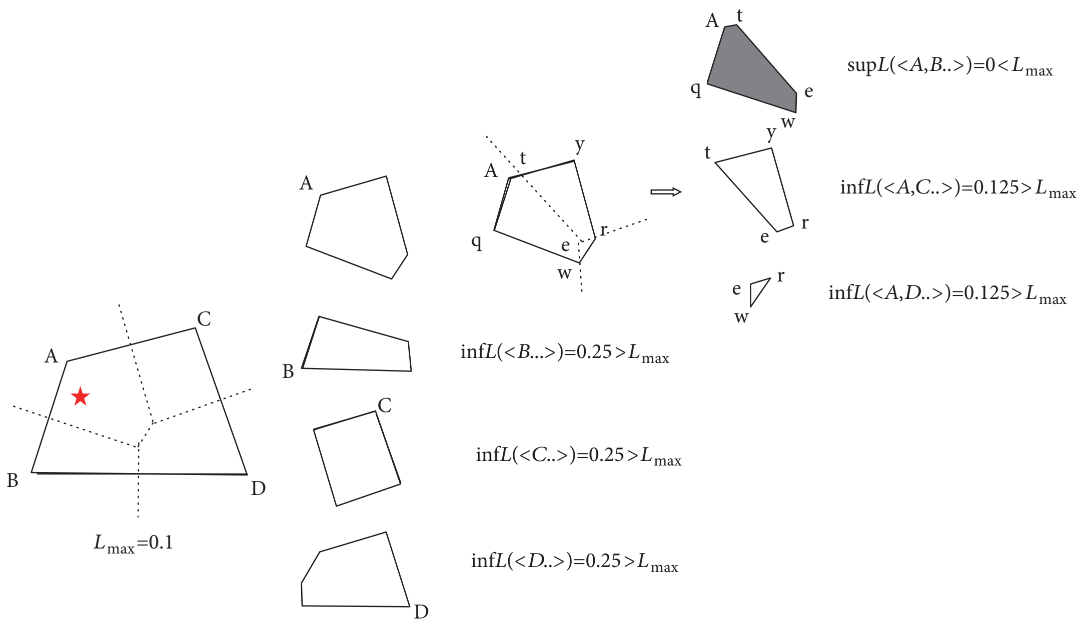
5. Experimental Evaluation

In all experiments, we use the real Harbin Station (45.768038, 126.644593) as the real location l and query the banks within 400 meters. The real result can be obtained by calling Baidu Maps. For the sake of simplicity, we only take the top 10 points of interest into consideration and regard the others as the extra ones. We ran our experiments on a desktop computer with an Intel Core i5-7200 2.50 GHz processor and 8 GB RAM. The real query result and the ranking are as follows.

Since the triangle made of (110, 110), (120, 120), (130, 130) has the same shape as the triangle made of (10, 10), (20, 20), (10, 30), we take the common prefix away to get

```
Input: Queue $q$; Delaunay triangulation $dt$; Set of Candidate $Candy$;
Output: set of Candidate $Candy$;
1. for each $x \in q$ do
2.     for each Simplex $t \in dt$ do
3.     if $t.contains(x)$ then
4.         l=t.vertices;
5.         for each $v \in l$ do
6.             if $v \notin Candy$ then
7.                 Candy.add(v);
8. $Sort(Candy)$ with ranking;
9. return $Candy$;
```

Algorithm 5: Generate the candidate of a queue.



A
B
C
D
$L_{\max}=0.1$
A
B
C
D
A
t
y
q
e
r
w
$\Longrightarrow$
A
t
q
e
w
$\sup L(<A,B..>)=0<L_{\max}$
t
y
e
r
$\inf L(<A,C..>)=0.125>L_{\max}$
e
r
w
$\inf L(<A,D..>)=0.125>L_{\max}$
$\inf L(<B...>)=0.25>L_{\max}$
$\inf L(<C..>)=0.25>L_{\max}$
$\inf L(<D..>)=0.25>L_{\max}$

Figure 8: Calculate the obfuscated region.

a smaller coordinate value for computing triangulation conveniently. We change the top one POI (14097372.321867, 5711495.970734) to (372.321867,495.970734) and then do the same thing for the others as in Table 1. The change of coordinates will also induce the change of the distance between the real location and each query result. We regard the distance between the last one POI transformed (710.269230,3.976522) and the real location transformed (433.443102926,417.242938906) as the query radius, which is $r_{max} = 500$.

5.1. Performance. For the first situation which is $loss(P_n, P_r) = 0$, as defined in Section 3, the last two POI are interchangeable. We got the ultimately obfuscated region in terms of that. In order to realize the tolerable quality of services ($loss(P_n, P_r) \leq L_{max}$), given L_{max}, the key question is how to get all the ranking results. A useful lemma combining the classical triangulation is shown as follows.

Table 1

Rank	Geographic	Horizontal
1	126.644040,45.768543	14097372.321867,5711495.970734
2	126.643815,45.768693	14097347.451385,5711519.173524
3	126.643240,45.769379	14097283.918655,5711626.604204
4	126.646863,45.766884	14097684.462267,5711240.031558
5	126.647147,45.769253	14097716.036524,5711617.643764
6	126.648564,45.768321	14097872.775613,5711473.096577
7	126.648580,45.768274	14097874.543330,5711465.660886
8	126.648690,45.767788	14097886.686469,5711388.631129
9	126.641381,45.766047	14097078.182499,5711090.967867
10	126.647097,45.765396	14097710.269230,5711003.976522

Lemma 1. *If the ranking of the current generator (vertex) v is r, C represents the set of vertices within the triangulations that*

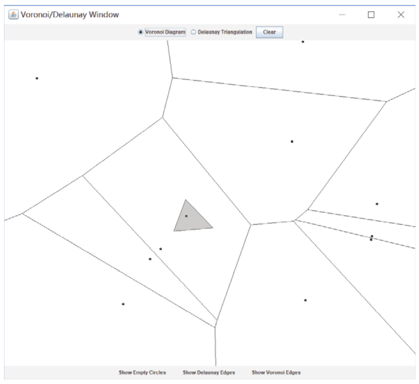


Figure 9: Obfuscated region with nonlossy service quality ($\lambda = 1, a = 2$).

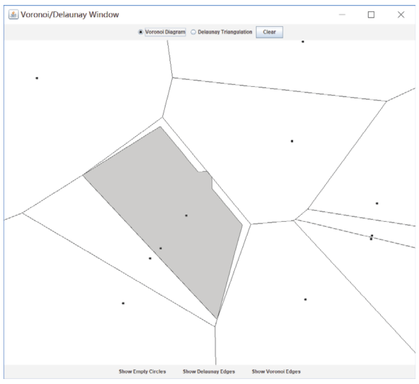


Figure 10: Obfuscated region with tolerance ($\lambda = 1, a = 2, L_{max} = 0.2$).

contain r. Then all the possible vertices in ranking $r + 1$ are expressed as $v' \in C$ $(v' \neq v)$.

Proof. The Delaunay triangulation (TIN) gathers the 3 nearest neighbors, and each generator (vertex) has a public edge with the others in the Voronoi diagram [33]. Since $dis(v', v) \forall v \in C < dis(v', others)$, the vertices which can be ranking in $r + 1$ must have a public edge with the current generator (vertex). □

As shown in Figure 9, any points in the gray area satisfy the nonlossy service quality. The user's real location is protected while receiving the highest quality of service. We utilize the theory of the Voronoi triangulations instead of simple enumeration to slump the time complexity of the ranking calculation. As shown in Figure 10, we set the weight of rank and the maximal tolerance as $a = 2$ and $L_{max} = 0.2$. The distance between each point of the obfuscated region and any one query result is within the maximum query ranges, which can be expressed as $dis(v, l') < r_{max}$.

5.2. Effect of L_{max}. We studied the scalability of our method by varying L_{max} in the range of 0.1 to 0.25. The weighting parameter a was 2. Figure 11 presents the obfuscated region when L_{max} increases from 0.1 to 0.25. As can be observed, the obfuscated region increases constantly as L_{max} increases, which realizes more protection of the user's location. It is slow growth when L_{max} increases from 0.1 to 0.2, but when we set $L_{max} = 0.25$, the obfuscated region nearly doubles on account of the change of the higher rankings.

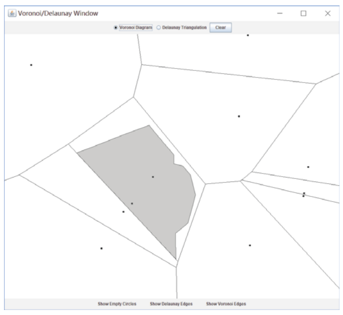
5.3. Effect of the Weighting Parameter a. We studied the scalability of our method by varying a in the range of 2 to 4 and λ in the range of 1 to 3. L_{max} was 0.2. Figure 12 presents the obfuscated region when the weighting parameter $(\lambda a^{(} - 1))$ increases from 1/2 to 3/4. As can be observed, the obfuscated region shrinks constantly as $\lambda a^{(} - 1)$ increases, which realizes better quality of service. It is significant change when weighting parameter increases from 1/2 to 2/3, and there is no obvious change from 2/3 to 3/4.

5.4. Errors of Perturbation. We now compare the perturbation errors of our scheme with the Laplace noises from 10 experiments. We set the parameter of global sensitivity as Δf=300 and vary ϵ from 0.5 to 1 and L_{max} from 0 to 0.15. Figure 13 depicts the comparison results and we can see that ϵ decreasing and L_{max} increasing can lead to the perturbation error increases. Besides, our scheme achieves fewer errors than the Laplace noise.

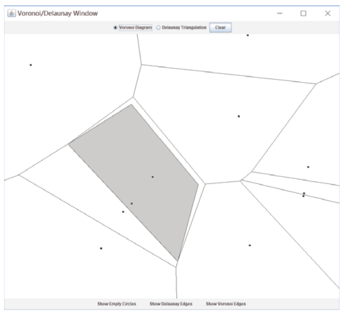
5.5. Comparison of Ranking Calculating Time. We then look at the ranking calculating time of our enumeration algorithms: enumeration with pruning and enumeration with the Voronoi diagram. We set $L_{max} = 0.1$ and perform 10 experiments. The results are shown in Figure 14. We can see that the time cost of enumeration with the Voronoi diagram is approximately 30% of enumeration with pruning. Experiments prove the enumeration with the Voronoi diagram can effectively reduce time complexity.

6. Conclusion

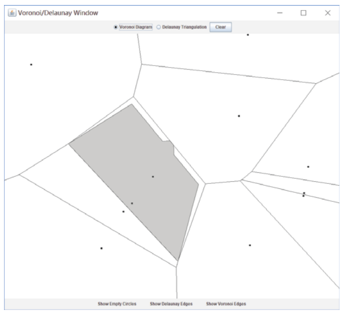
In this paper, we propose a novel location privacy protection model based on the loss of service quality. A trusted third party (TTP) is added between the user and the server, for collecting the user's real location and then sending to the LBS



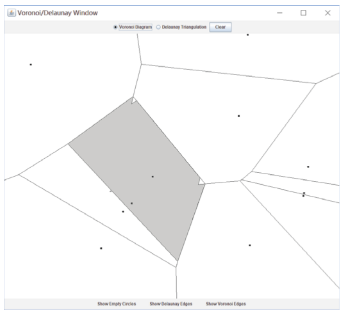
(a) $L_{max} = 0.1$



(b) $L_{max} = 0.15$

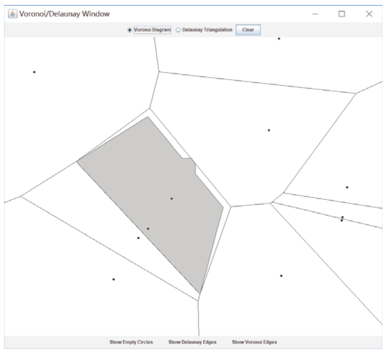


(c) $L_{max} = 0.2$

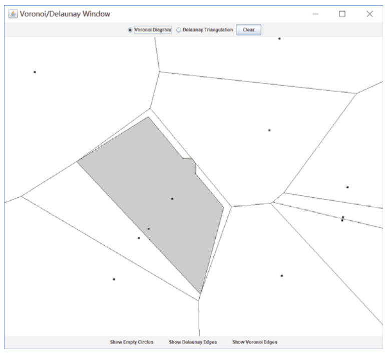


(d) $L_{max} = 0.25$

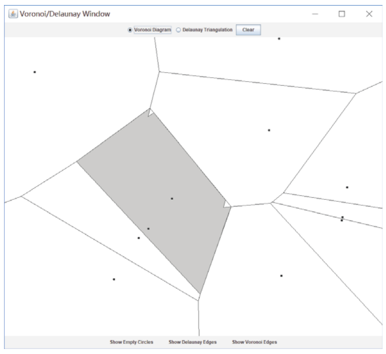
Figure 11: Effect of L_{max}.



(a) $\lambda = 1, a = 2$



(b) $\lambda = 2, a = 3$



(c) $\lambda = 3, a = 4$

Figure 12: Effect of the weighting parameter a and λ.

server with the disturbed one. The model allows a user to express his/her requirement of service quality by specifying a maximum service quality loss L_{max}, which the user would tolerate. The loss of service quality L_{max} can also be set to 0. Find all possible rankings under satisfying constraints to get the final obfuscated region, and then select one point at random as the obfuscated location. In order to ensure the excellent service, L_{max} is usually set to a smaller one.

In this paper, we only see the user's location for privacy and a novel strategy based on the Voronoi diagram is used to generate the noisy location in order to improve the service quality. Since the query privacy is also a key privacy concern, the query interests incur potential damage to personal privacy and even to individual safety. How to ensure the query privacy is another area we would like to investigate further.

Data Availability

The data used in our paper is just longitude and latitude of points of interest (POI), and these are public and can be obtained by anyone. And the third party of our work is Baidu Maps API, which is also public.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This article is partly supported by the National Natural Science Foundation of China under Grant No. 61370084

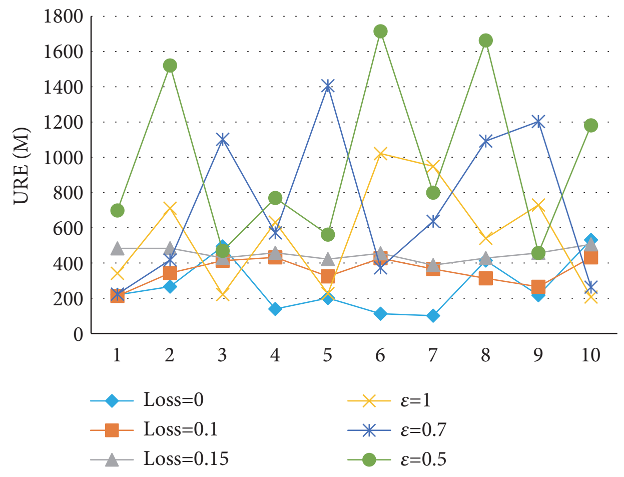
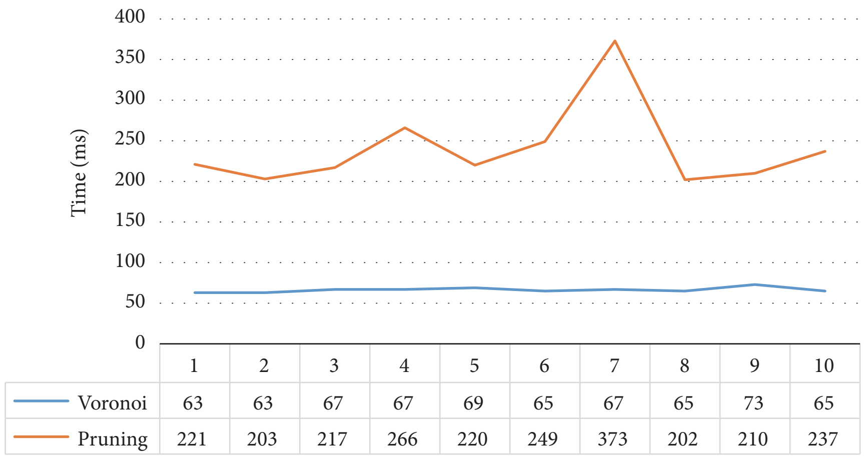


FIGURE 13: Errors of perturbation.



	1	2	3	4	5	6	7	8	9	10
Voronoi	63	63	67	67	69	65	67	65	73	65
Pruning	221	203	217	266	220	249	373	202	210	237

FIGURE 14: Comparison of ranking calculating time.

and No. 61872105, the Experimental Verification of the Basic Commonness and the Key Technical Standards of the Industrial Internet Network Architecture, the Key Technology of Home-Based Care Service System (2016RAXXJ013), and Fundamental Research Funds for the Central Universities (Grant No. 3072019CF0602).

References

[1] Q. Han, S. Liang, and H. Zhang, "Mobile cloud sensing, big data, and 5G networks make an intelligent and smart world," *IEEE Network*, vol. 29, no. 2, pp. 40–45, 2015.

[2] K. Zhang, Q. Han, Z. Cai, and G. Yin, "RiPPAS: a ring-based privacy-preserving aggregation scheme in wireless sensor networks," *Sensors*, vol. 17, no. 2, p. 300, 2017.

[3] X. Zheng, Z. Cai, and Y. Li, "Data linkage in smart internet of things systems: a consideration from a privacy perspective," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 55–61, 2018.

[4] M. F. Mokbel, "Privacy in location-based services: state-of-the-art and research directions," in *Proceedings of the 8th International Conference on Mobile Data Management (MDM)*, p. 228, IEEE, Mannheim, Germany, 2007.

[5] Z. Cai and Z. He, "Trading private range counting over big iot data," in *Proceedings of the 39th IEEE International Conference on Distributed Computing Systems*, 2019.

[6] Y. Liang, Z. Cai, J. Yu, Q. Han, and Y. Li, "Deep learning based inference of private information using embedded sensors in smart devices," *IEEE Network*, vol. 32, no. 4, pp. 8–14, 2018.

[7] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 577–590, 2018.

[8] Z. Cai, X. Zheng, and J. Yu, "A differential-private framework for urban traffic flows estimation via taxi companies," *IEEE Transactions on Industrial Informatics*, 2019.

[9] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez, "Show me how you move and I will tell you who you are," *Transactions on Data Privacy*, vol. 4, no. 2, pp. 103–126, 2011.

[10] J. Krumm, "Inference attacks on location tracks," in *Proceedings of the Pervasive Computing, 5th International Conference (PERVASIVE)*, pp. 127–143, Toronto , Canada, 2007.

[11] Y. Matsuo, N. Okazaki, K. Izumi et al., "Inferring long-term user properties based on users location history," in *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2159–2165, Hyderabad, India, 2007.

[12] Z. Cai and X. Zheng, “A private and efficient mechanism for data uploading in smart cyber-physical systems,” *IEEE Transactions on Network Science and Engineering*, 2018.

[13] Y. Huo, X. Fan, L. Ma, X. Cheng, Z. Tian, and D. Chen, “Secure communications in tiered 5G wireless networks with cooperative jamming,” *IEEE Transactions on Wireless Communications*, vol. 18, no. 6, pp. 3265–3280, 2019.

[14] A. R. Beresford and F. Stajano, “Location privacy in pervasive computing,” *IEEE Pervasive Computing*, vol. 2, no. 1, pp. 46–55, 2003.

[15] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan, “Private queries in location based services: anonymizers are not necessary,” in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pp. 121–132, Vancouver, BC, Canada, 2008.

[16] P. Samarati and L. Sweeney, “Generalizing data to provide anonymity when disclosing information (abstract),” in *Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, p. 188, Seattle, Wash, USA, 1998.

[17] H. Kido, Y. Yanagisawa, and T. Satoh, “Protection of location privacy using dummies for location-based services,” in *Proceedings of the 21st International Conference on Data Engineering Workshops (ICDEW)*, p. 1248, Tokyo, Japan, 2005.

[18] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, “Geo-indistinguishability: differential privacy for location-based systems,” in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 901–914, ACM, Berlin, Germany, 2013.

[19] C. Dwork, “Differential privacy,” in *Proceedings of the in Automata, Languages and Programming, 33rd International Colloquium (ICALP)*, pp. 1–12, Venice , Italy, 2006.

[20] S. Oya, C. Troncoso, and F. Pérez-González, “Is geo-indistinguishability what you are looking for?” in *Proceedings of the 2017 on Workshop on Privacy in the Electronic Society*, pp. 137–140, Dallas, Tex, USA, 2017.

[21] O. Abul, F. Bonchi, and M. Nanni, “Never walk alone: uncertainty for anonymity in moving objects databases,” in *Proceedings of the IEEE 24th International Conference on Data Engineering (ICDE)*, pp. 376–385, IEEE, Cancun, Mexico, 2008.

[22] Q. Han, D. Lu, K. Zhang, X. Du, and M. Guizani, “Lclean: a plausible approach to individual trajectory data sanitization,” *IEEE Access*, vol. 6, pp. 30110–30116, 2018.

[23] Q. Han, B. Shao, L. Li, Z. Ma, H. Zhang, and X. Du, “Publishing histograms with outliers under data differential privacy,” *Security and Communication Networks*, vol. 9, no. 14, pp. 2313–2322, 2016.

[24] X. Zheng, Z. Cai, J. Li, and H. Gao, “Location-privacy-aware review publication mechanism for local business service systems,” in *Proceedings of the IEEE INFOCOM - Conference on Computer Communications*, pp. 1–9, IEEE, Atlanta, Ga, USA, 2017.

[25] C. Dwork, F. McSherry, K. Nissim, and A. D. Smith, “Calibrating noise to sensitivity in private data analysis,” in *Proceedings of the 3rd Theory of Cryptography Conference (TCC)*, pp. 265–284, Springer, New York, NY, USA, 2006.

[26] Q. Han, Q. Chen, L. Zhang, and K. Zhang, “HRR: a data cleaning approach preserving local differential privacy,” *International Journal of Distributed Sensor Networks*, vol. 14, no. 12, 2018.

[27] Q. Han, Z. Xiong, and K. Zhang, “Research on trajectory data releasing method via differential privacy based on spatial partition,” *Security and Communication Networks*, vol. 2018, Article ID 4248092, 14 pages, 2018.

[28] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, “Local privacy and statistical minimax rates,” in *Proceedings of the 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, p. 1592, Allerton Park & Retreat Center, Monticello, IL, USA, 2013.

[29] https://www.internetmarketingninjas.com/.

[30] https://en.wikipedia.org/wiki/Voronoi_diagram/.

[31] http://lbsyun.baidu.com/.

[32] J. A. O’Keefe, “The universal transverse mercator grid and projection,” *The Professional Geographer*, vol. 4, no. 5, pp. 19–24, 1952.

[33] M. Mostafavi, G. Abolfazl, Christopher., and D. Maciej, “Delete and insert operations in voronoi/delaunay methods and applications,” *Computers and Geosciences*, vol. 29, no. 4, pp. 523–530, 2003.