A Location Privacy Preserving Method Based on Sensitive Diversity for LBS

Changli Zhou[1,*], Chunguang Ma[1,*], Songtao Yang[1,2], Peng Wu[1], and Linlin Liu[3]

[1] Harbin Engineering University, Harbin City 150001, China
[2] Jia Mu Si University, Jiamusi City 154007, China
[3] Harbin Crystal Commercial Photography Co. Ltd, Harbin City 150001, China
zhouchangli888@gmail.com, machunguang@hrbeu.edu.cn

Abstract. A user's staying points in her trajectory have semantic association with privacy, such as she stays at a hospital. Staying at a sensitive place, a user may have privacy exposure risks when she gets location based service (LBS). Constructing cloaking regions and using fake locations are common methods. But if regions and fake positions are still in the sensitive area, it is vulnerable to lead location privacy exposure. We propose an anchor generating method based on sensitive places diversity. According to the visiting number and peak time of users, sensitive places are chosen to form a diversity zone, its centroid is taken as the anchor location which increases a user's location diversity. Based on the anchor, a query algorithm for places of interest (POIs) is proposed, and precise results can be deduced with the anchor instead of sending users' actual location to LBS server. The experiments show that our method achieves a tradeoff between QoS and privacy preserving, and it has a good working performance.

1 Introduction

Location Based Service (LBS) brings convenience to people's lives, at the same time, it also poses a risk of location privacy leakage. Location based query is a widely used LBS, a user sends a query request with her current location to LBS provider (LSP) to get places of interest (POIs). Such as "find the *K* nearest neighbor restaurants around me" or "find all the restaurants in the range of *R* kilometers", the former one is called *K*NN query and latter one is range query. Due to the spatial and temporal relevance, an exposure of location privacy may lead deeply privacy leakage, such as a user's home address, hobbies, health condition and so on. Location privacy is significantly important to us and should be protected carefully.

Places on a user's trajectory can be divided into two kinds: passing-by places and staying-at places. A mobile user issues LBS query with her current location at any time in a trajectory. A passing-by place has no relationship with a user, it only means a user has passed by a location without any semantic association. But a staying-at place, especially a sensitive place, has semantic association with a user staying at it, such as a user is staying at an infectious hospital.

* Corresponding author.

C.-H. Hsu et al. (Eds.): NPC 2014, LNCS 8707, pp. 409–422, 2014.

Location obfuscation is a general protecting method for location privacy preserving. Such as constructing cloaking region to achieve *k*-anonymity[1,2,3], as shown in Fig 1, user C sends her actual location to an anonymous server (AS), then AS expands her actual location to a rectangle R_2 including 2 other users, and R_2 will be sent to LSP for POIs instead of her actual location. But there is a problem, if the cloaking region is in a sensitive area, such as dash line rectangle in Fig.2. A query is sent with R_2 means the user is in a hospital. And when a user stays or moves a short distance in a sensitive area, all her cloaking regions may be included in it. Location diversity is a solution that requires users in a cloaking region to appear in diverse places, but that may lead a large cloaking region, such as R_3.

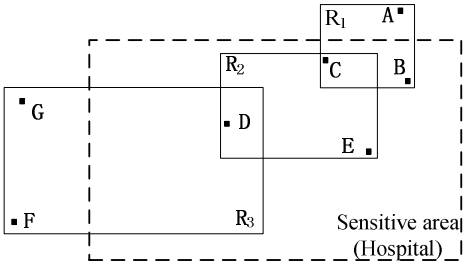

Fig. 1. Cloaking regions

Another protecting method is using fake locations[6,7], that is sending an actual location accompanied with some fake locations, and all the locations will be used in query operations, that brings too much burden to LSP. Then query methods with significant object[8] or anchor[9] are proposed, they have more improvements and more precise query results. Especially, SpaceTwist[9] is an effective method to get *K*NN POIs without providing a user's actual location to LSP. But these methods have the same drawback, which is if the fake locations or anchors are still picked in a sensitive area, location privacy of a user will be leaked anyway.

Staying at a sensitive place causes a semantic association with a user, continuous sensitive places lead to deep-going leakages[10,11]. We focus on the privacy preserving when a user is staying at or moving short distance around a sensitive place. The contents and contributions of this paper are as follows:

1).We propose a location privacy preserving method based on sensitive places diversity when a user is staying at a sensitive place. A center server (CS) generates a diversity anchor for a user. The diversity anchor is used to replace a user's actual location. CS sends a query with the anchor. The diversity anchor is in the overlap area of several sensitive places, which increases uncertainty of a user's actual location.

2).We propose a query algorithm with the diversity anchor. In a query request, a diversity anchor is sent to LSP instead of a user's actual location. LSP takes the anchor as a centroid and returns a candidate POIs set to CS, and CS can deduce precise result of *K*NN POIs for a user. Without providing any user's actual location, our algorithm achieves location privacy preserving and gets precise *K*NN POIs for a user.

2 Related Works

In order to achieve location privacy preserving, a user obscures her actual location before getting LBS. Gruteser et al[1] brought in *k*-anonymous idea from database for LBS privacy preserving. Mokbel et al[12] proposed an architecture with center server(CS), CS is between users and LSP, most of the CSs are credible. CS cloaks a user's actual location and returns refined results. Chow and Mokbel[13,14] proposed a P2P architecture without CS, it removes bottleneck when CS faces lots of users.

Anonymity is achieved by these methods, but if users crowd together in a place, cloaking regions may still in a small area, in extreme case they are at the same spot. To solve this problem, Bamba et al[4] introduced *l*-diversity idea from data publication into location anonymity, they proposed a cloaking method which satisfies location diversity. Xue et al[17] proposed a location diversity method to ensure each query can be associated with at least *l* different semantic places. Xu et al[5] proposed an anonymous cell with diversity roads. Yang et al[18] proposed cloaking cycle and forest which include diversity roads to ensure that a user locates at diversity roads equally in a cloaking cycle. Meng et al[11] proposed sensitive trajectory location protection method in data publication. Liu[19] gives query *l*-diversity in location privacy preserving for the first time.

Using fake locations is another way to achieve protection. A general method is sending several fake locations in order to obscure a user's actual location [6-9]. A user sends a fake location in SpaceTwist [9], which is called “anchor”, to LSP and the user deduces POIs result according to the returned candidate set. The main procedure is as follows:

As shown in Fig. 2, a solid “•” denotes a user's actual location, “×” denotes an anchor. A user sends a query with the anchor to LSP, LSP performs INN (incremental nearest neighbor) query to get POIs candidate set and then sends it to the user gradually. Firstly, LSP takes the anchor as the centriod of supply space to search POIs. When a POI is found in Fig.2(b), the supply space expends and the demand space centred with user's location shrinks. As POIs are found gradually, SpaceTwist terminates when the supply space covers the demand space. Meng[15] and Gong[16] have proposed improvement to make SpaceTwist achieve *k*-anonymity respectively.

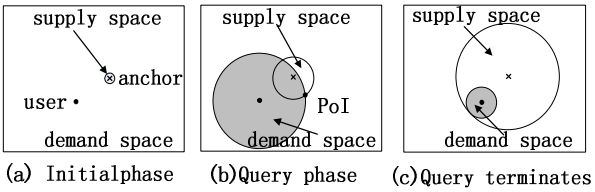

Fig. 2. SpaceTwist processing procedure

Both cloaking region and anchor will cause privacy leakage when they are still in a sensitive place. In this paper, we use an anchor referring to SpaceTwist, and ensure a user at sensitive place to pick the anchor with location diversity. Based on the anchor, we propose a query algorithm to get precise *K*NN POIs result for a user.

3 System Architecture

We pick the architecture with a CS, CS is between users and LSP, as shown in Fig. 3. A user with a GPS sensor of her intelligent terminal sends her location and query to CS. CS computes an anchor and sends anonymous query with the anchor to LSP. LSP performs INN search in its database according to the anchor location and returns POIs candidate set to CS. CS deduces precise results to the user.

Definition 1. There are 3 entity sets $<\mathrm{U},\mathrm{CS},\mathrm{LSP}>$, $u_k \in \mathrm{U}$ represents an energy constrained mobile user. $CS_i \in \mathrm{CS}$ is a central server, deployed at crowded location, it has stronger abilities. LSP is an LBS provider, which is powerful in energy and processing, it stores all POIs in its database. CS is credible, users and LSP may be not.

Definition 2. A user's query $<u_k, loc_{uk}, l, C, R>$, u_k is her identity, loc_{uk} is an actual location, l is sensitive diversity degree, C and R are her query request content and personal requirement in the query respectively.

Definition 3. CS sends a piece of query $<CS_i, loc_{anchor}, C, \beta>$, CS_i is identity of a CS and loc_{anchor} is the anchor location which is computed and satisfied with location diversity, β is the number of POIs returned from LSP each time.

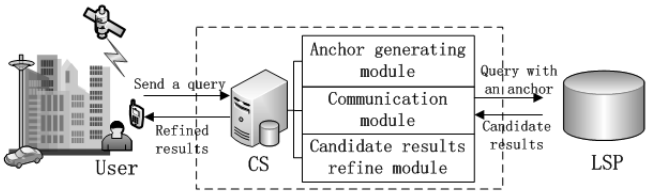

Fig. 3. System architecture

Definition 4. POIs are denoted as $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, ..., \mathcal{P}_n\}$, $\mathcal{P}_i \in \mathcal{P}$ is a POI or a sensitive place. POIs also have semantic association with users, so we usually consider some POIs as sensitive places.

4 Location Privacy Preserving Method

Our method includes two main phases: the CS generates a diverse anchor for the user who is at a sensitive place, and query for *K*NN POIs with the anchor. The first phase contains sensitive location definition method based on users visiting frequency characteristics, and the anchor generating method is based on sensitive locaiton diversity. The second phase presents the query algorithm with a diversity anchor.

4.1 Diversity Anchor Generating Phase

We assign different sensitive weights based on users visiting number and visiting time period firstly. The sensitive weights are used to generate a diversity anchor then.

4.1.1 Sensitive Location Definition

Visiting number and peak visiting time period of a place reflect a sociality of a kind of people. When the users are staying at the place, the semantic associations will lead a privacy leakage of these users. For example, the visiting users to a place becomes more in every weekday morning, it may be a company rather than a bar, a user stays at this place may expose her working place. A place is often visited at night, it may be a bar rather than a hospital. Nearly all sensitive places have bigger visiting numbers and regular peak visiting time. These may lead a correlation with a category of places, so we take visiting number and peak visiting time as main factors.

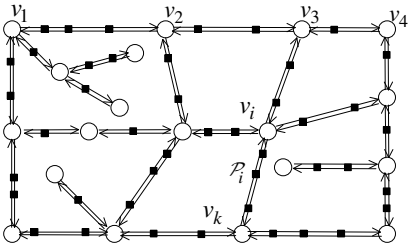

Fig. 4. Road networks with POIs

As the sensitive places are distributed in road networks, a user always finds a path to reach a sensitive place. So when we discuss users visiting number, we consider sensitive places (or POIs) are on the edge of the road graph. We define a directed graph of road networks as $G = (\mathbf{V},\mathbf{E})$, $\mathbf{V}$ is a set of vertexes, each $v_i \in \mathbf{V}$ has a visiting weight $\mathcal{R}(v_i) = \lambda_i$. $\mathbf{E}$ is a set of edges, $e_{ik} \in \mathbf{E}$ is a directed edge between v_i and v_k. If there is no other vertex $v_x \in \mathbf{V} / \{v_i, v_k\}$ between v_i and v_k, a road directly connects v_i and v_k. Users arrive from v_i to v_k follows Poisson process with arrival rate $\lambda_{ik} > 0$, and $e_{ik} = \lambda_{ik}$, or else $e_{ik} = 0$. So we define visiting weight of a vertex v_i:

$$\mathcal{R}(v_i) = \lambda_i = \lambda_i^{'} + \sum_{v_j \in \mathbf{V}, k \neq i} e_{ki} = \lambda_i^{'} + \sum_{v_j \in \mathbf{V}, k \neq i} \lambda_{ki} \tag{1}$$

$\lambda i^{'}$ is a accumulation of user arrival rate who doesn't start from a vertex. Suppose a user chooses each outgoing edge of a vertex with equal probability, each outgoing edge has a visiting weight $\mathcal{R}(v_i) / \deg_{out}(v_i)$, $\deg_{out}(v_i)$ is the outgoing degree. A road segment with two vertexes v_i and v_k has a weight $\mathcal{M}$ in the Formula (2). As shown in Fig.4, black square points are denoted as sensitive places. As we known, a user doesn't stays at each places in a road segment $v_i v_k$, she may only stay at one place according to her destination.

$$\mathcal{M} = [\mathcal{R}(v_i) / \deg_{out}(v_i)] + [\mathcal{R}(v_k) / \deg_{out}(v_k)] \tag{2}$$

Suppose a place $\mathcal{P}_i$ on $v_i v_k$ has n users passed by in a certain time period of a day and the probability of staying-at users is p, so the users staying at a place $\mathcal{P}_i$ on $v_i v_k$ follows Poisson process with an arrival rate $\mu_i = np$. The probability of staying-at number X of users when X is greater than a threshold $\mathrm{X_T}$ is:

$$P(X > \mathrm{X_T}) = 1 - P(X \leq \mathrm{X_T}) = 1 - \frac{e^{-\mu}}{0!} - \frac{\mu e^{-\mu}}{1!} - \ldots - \frac{\mu^{\mathrm{X_T}} e^{-\mu}}{\mathrm{X_T}!} \tag{3}$$

So each place can be assigned with the weight as:

$$\mathcal{R} = \mathcal{M} \cdot P(X > X_T) \tag{4}$$

We choose typical time periods of a day, such as rush hour, leisure time and so on, to get a sensitive weights sequence of a place $\mathcal{R}(\mathcal{P}_i) = (\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3, \ldots, \mathcal{R}_n)$, we can get its peak visiting time periods of a day. The average value $\overline{\mathcal{R}(\mathcal{P}_i)}$ in Formula (5) reflects average visiting number of a place.

$$\overline{\mathcal{R}(\mathcal{P}_i)} = \sum_{i=1}^{n} \mathcal{R}_i \Big/ n \tag{5}$$

If a place satisfies $\overline{\mathcal{R}(\mathcal{P}_i)} > R_T$, we call it a sensitive place, R_T is a sensitive threshold. The sensitive weights are used to generate diversity anchor in next section.

Anchor generating based on sensitive location diversity

In this section, we pick a user's neighbor sensitive places to form a diversity zone, the anchor is generated at the centroid of the zone, a user querying with the anchor improves the probability of staying at different sensitive places.

When CS chooses neighbor sensitive places for a user, we divide neighbor sensitive places into 3 categories:

A. Disparate places, this kind of places have disparate peak visiting time period, a user choose this place may lead severely uneven distributing probability of each sensitive places for a user, such as a hospital and a bar, so CS excludes these places.

B. High correlation places, this kind of places do not only have similar peak visiting time period but also shows a linear correlation with the sensitive place which the user is staying at. These places may be the same kind neighbor places, such as two neighbor bars. For achieving diversity, CS excludes these places.

C. Similar places, this kind of places have similar peak visiting time period but they are not the same places, choosing this kind of places ensures sensitive diversity.

There are other measures to pick diversity places, we focus on user visiting number and its variation tendency according to the sensitive weight sequence of a place $\mathcal{R}(\mathcal{P}_i) = (\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3, \ldots, \mathcal{R}_n)$, which we have discussed below Formula (4).

CS has the sensitive weight sequences of all the POIs in its coverage area, one of the sequences of a place $\mathcal{P}_i$ is denoted as $\mathcal{R}(\mathcal{P}_i) = (\mathcal{R}_1^i, \mathcal{R}_2^i, \mathcal{R}_3^i, \ldots, \mathcal{R}_n^i)$, each $\mathcal{R}_1^i \in \mathcal{R}(\mathcal{P}_i)$ at different time periods is computed by Formula (4). Suppose a user is staying at $\mathcal{P}_i$, and $\mathcal{P}_k$ is one of its neighbor sensitive places. CS compares the sequence $\mathcal{R}(\mathcal{P}_i)$ to all the neighbor sensitive places $\mathcal{R}(\mathcal{P}_k)$ and excludes the ones belonging to category A. We use cosine similarity to achieve this goal, in Formula (6), since cosine similarity can reflect the tendency similarity of two data sequences, $sim(\mathcal{P}_i, \mathcal{P}_k) \in [0,1]$, low similarity means a disparate place.

$$sim(\mathcal{P}_i, \mathcal{P}_k) = \frac{\mathcal{R}(\mathcal{P}_i) \cdot \mathcal{R}(\mathcal{P}_k)}{\|\mathcal{R}(\mathcal{P}_i)\| \cdot \|\mathcal{R}(\mathcal{P}_k)\|} = \frac{\sum_{j=1}^{n} \mathcal{R}_j^i \times \mathcal{R}_j^k}{\sqrt{\sum_{j=1}^{n} (\mathcal{R}_j^i)^2} \times \sqrt{\sum_{j=1}^{n} (\mathcal{R}_j^k)^2}} \tag{6}$$

Formula (6) only filters the disparate places. If two sequences of $\mathcal{P}_i$ and $\mathcal{P}_k$ show similar tendency, such as 2 simple examples (2000, 400, 100) and (1000, 200, 50), they have similar variation tendency, and shows linear similarity, these may belong to category B. We exclude these places to guarantee sensitive diversity. We use Pearson correlation coefficient to achieve this goal, which represents the linearly dependent of two data sequences.

$$r(\mathcal{P}_i,\mathcal{P}_k)=\frac{1}{n-1}\sum_{j=1}^{n}\left(\frac{\mathcal{R}_j^i-\overline{\mathcal{R}_i}}{s_{\mathcal{R}_i}}\right)\left(\frac{\mathcal{R}_j^k-\overline{\mathcal{R}_k}}{s_{\mathcal{R}_k}}\right) \tag{7}$$

As shown in Formula (7), $\overline{\mathcal{R}_i}$ 和 $S_{\mathcal{R}i}$ are mean value and standard deviation respectively. The more $|r|$ approaches 1, the higher linearly dependent is. We exclude places of category B with high $|r|$. There is no negative correlation (r<0) after the filter of Formula (6).

$$Dist(\mathcal{P}_i,\mathcal{P}_k)=\sqrt{\sum_{j=1}^{n}(\mathcal{R}_j^i-\mathcal{R}_j^k)} \tag{8}$$

CS filters disparate places and high correlation places by Formula (6) and (7), the remaining places satisfy sensitive diversity and refrains from inferring attack according to peak visiting time period difference. We rank the remaining candidate places according to similar degree, as defined in Formula (8), CS chooses better places to form a diversity zone according to diversity degree. We use Euclidean distance to estimate the similar degree in the candidate set. The greater Euclidean distance is, the higher diversity degree of a neighbor sensitive place is.

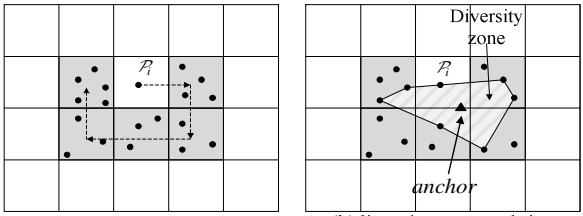

Fig. 5. Diversity zone and anchor generating

As the area is divided into grids by default, when CS receives a query from a user staying at $\mathcal{P}_i$ $<u_k, loc_{uk}, l, C, R>$, it picks a neighbor grid randomly, as shown in Fig.5 (a), and clockwise get all the sensitive places in its neighbor grids, all the grids are in an angle range of 180° from the first grid, there is an angle limit because if the other sensitive places surround $\mathcal{P}_i$, $\mathcal{P}_i$ will be the sensitive place where the user is staying at. CS compares each sensitive place with $\mathcal{P}_i$ using Formula (6) and (7), filters disparate places and high correlation places, and ranks the remaining places according to Formula (8). Finally, CS chooses l sensitive places to form a diversity zone and takes its centroid as the anchor location, l is sensitive diversity degree defined by the user in query request. The Algorithm is as follows:

Algorithm 1. Diversity zone and anchor generating

1. **Procedure :** CS receives a query request $<u_k, loc_{uk}, l, C, R>$ from a user at $\mathcal{P}_i$
2. generate a max heap W
3. randomly pick a neighbor grid, denote the vector from $\mathcal{P}_i$ to the grid as v_1
4. **while** $\theta(v_1, v_i) \le 180^\circ$ // v_i is the vector which u_k points to the ith neighbor grid
5. clockwise get all neighbor grids
6. $S \leftarrow$ all the sensitive places in these grids
7. **for** each $\mathcal{P}_k \in S$ **do**
8. compute $sim(\mathcal{P}_i, \mathcal{P}_k)$
9. **while** $sim(\mathcal{P}_i, \mathcal{P}_k) > \xi_s$ **do** // ξ_s is a threshold
10. compute $r(\mathcal{P}_i, \mathcal{P}_k)$
11. **if** $r(\mathcal{P}_i, \mathcal{P}_k) < \xi_r$ **then** // ξ_r is a threshold
12. compute $Dist(\mathcal{P}_i, \mathcal{P}_k)$
13. $W \leftarrow \mathcal{P}_k, Dist(\mathcal{P}_i, \mathcal{P}_k)$
14. **while** $|W| \ge l$ // satisfy sensitive l-diversity
15. connect the top l $\mathcal{P}_k \in W$ to form a $Zone_{\text{div}}$
16. $centroid \leftarrow$ compute the centroid of $Zone_{\text{div}}$ // take the centroid as an anchor for the user
17. **return** $centroid$
18. **End Procedure.**

In this section, we propose the picking method of sensitive diversity places according to user visiting number and its variation tendency. Then we use diversity places to form a diversity zone, the anchor is the centroid of the zone. CS uses this anchor to replace the user's actual location and issues users' query with the anchor. We can find that the anchor can be reused by other users in the sensitive places which form a diversity zone, the reuse decreases the overhead of CS.

4.2 Query Phase

In this phase, CS sends user's query request $<CS_i, loc_{anchor}, C, \beta>$ with a diversity anchor. When LSP receives a query request, it takes the anchor as a dimcenter and executes INN search. LSP returns the POIs candidate set gradually to CS. CS performs Algorithm 2 to deduce precise KNN PoIs for a user.

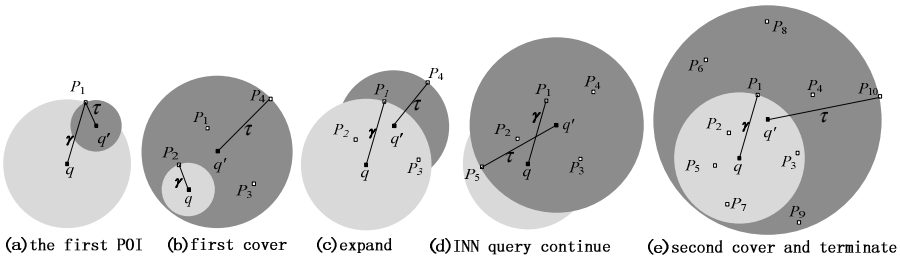

Fig. 6. K nearest neighbor PoIs query for a user

As show in Fig.6(a), a user locates at q and q' is the diversity anchor, when the first POI is found, supply space (the dark grey cycle) expands and demand space (light grey cycle) shrinks. As POIs are found gradually, supply space covers demand space for the first time in Fig.6(b), K POIs are found around the anchor. Then demand space updates, containing K POIs in its cycle and keeps its radius unchanged after the expand, as shown in Fig.6(c) K=3. In Fig.6(d-e), query procedure continues until supply space covers demand space for the second time, K POIs are found around user. The algorithm running at CS end and referring to SpaceTwist is as follows:

Algorithm 2. CS performs the algorithm for KNN PoIs around a user at q

1. **Procedure :** K is defined by u_k, $q \leftarrow loc_{uk}$, $q' \leftarrow loc_{anchor}$, β is the package capacity of PoIs returned from LSP
2. CS generates a max heap W_K
3. insert K pairs of $< NULL, \infty >$ into W_K
4. $\gamma \leftarrow$ the top distance in W_K // initialize demand space
5. $\tau \leftarrow 0$ // initialize supply space
6. send INN query to LSP with diversity anchor q'
7. **while** $\gamma + dist(q, q') > \tau$ **do**
8. $S \leftarrow$ get next package of PoIs from LSP
9. $\tau \leftarrow$ get the maximum $dist(q', \mathcal{P}_x)$ in S // update supply space
10. **for** each $\mathcal{P}_w \in S$ **do**
11. **if** $dist(q, \mathcal{P}_w) < \gamma$ **then**
12. $W_K \leftarrow < \mathcal{P}_w, dist(q, \mathcal{P}_w) >$
13. $\gamma \leftarrow dist(q, \mathcal{P}_w)$
14. $\gamma \leftarrow$ get $dist(q, \mathcal{P}_K)$ in W_K // update demand space
15. **while** $\gamma + dist(q, q') > \tau$ **do**
16. $S \leftarrow$ get next package of PoIs from LSP
17. $\tau \leftarrow$ get the maximum $dist(q', \mathcal{P}_u)$ in S // expand supply space gradually
18. **if** $dist(q, \mathcal{P}_u) < \gamma$ **then**
19. $W_K \leftarrow < \mathcal{P}_h, dist(q, \mathcal{P}_u) >$
20. terminate INN query
21. **return** bottom K PoIs in W_K
22. **End Procedure.**

In our algorithm, demand space expands and covers at least K PoIs, which is the key point guarantees the user to get K PoIs around him nearly in 100% success rate. The query process will not terminate until supply space covers demand space again. As shown in Fig.6(e), LSP returns 10 PoIs in total. Alogrithm 2 picks K=3 PoIs $\{\mathcal{P}_2, \mathcal{P}_5, \mathcal{P}_7\}$ of them, the 3 POIs are around the user q, our algorithm is better than SpaceTwist. When we consider a user stay in a sensitive place, that means all the users are static or moves short distance, Algorithm 2 is snapshot query rather than continuous query, a user in a query procedure always uses one diversity anchor. As we known, a continuous query is composed of several snapshot queries, so Algorithm 2 is applicable for continuous query if continuous anchor sequence is generated. We will consider it in future work.

4.3 Performance Analysis

In this section, we will discuss security in the procedure of diversity anchor generating and querying with the anchor, then we analysis the algorithm complexity.

(1) Security analysis

An anchor is chosen in the overlap region of several sensitive places, it increases the probability of a user appearing in different sensitive places, the user's location semantic privacy is preserved. The diversity sensitive places are filtered by Formula (6), the disparate places are discarded to ensure the user is staying at each places with fequal opportunity. Formula (7) filters the sensitive places which may be the same to the one a user is staying at, such as a user is staying at a hospital, CS choose neighbor other hospital for her, which reduces the diversity. At last, CS picks l sensitive places in the remaining places to form a diversity zone, since the sensitive places is chosen from a randomly direction firstly and different users at the same sensitive place have different *l*-diversity degrees, so CS generates different anchors for users from the same place, that avoids inferring attacks which all the users using the same anchor are from the same sensitive place.

When CS generates an anchor according to l sensitive places around him, she is staying at each place with equal probability $p(x_i) = 1/l$, so the information entropy of querying with this anchor one time is:

$$H(q) = \sum_{i=1}^{l} p(x_i) \log \frac{1}{p(x_i)} = \sum_{i=1}^{l} \frac{1}{l} \log l = \log l \tag{9}$$

That is the maximum information entropy for a single time, an adversary is hard to correlate any anchor with a user at sensitive place.

(2) Complexity analysis of query algorithm

Algorithm 2 is running at CS end, it compares the returned POIs from LSP, and decides when to terminate the query process, as demand space expanded in Algorithm2 Line14, the query terminated time has set already, so the algorithm will not last long or loop over and over again. The time complexity depends on amount of POIs returned in two phases in Algorithm 2 Line 8 and 16, it is $O(|\mathcal{P}_1|+|\mathcal{P}_h|)$. When K=3, LSP has to return 10 POIs to get precise KNN around a user, it is a little more, but the searching time complexity is not large. In the other hand, it is a tradeoff between ensuring privacy preserving and query efficiency.

5 Experiments

In this section, we discuss 3 main indicators: anonymity success rate, data traffic and average response time. We do experiments on two different data sets to manifest the good performance of our method.

5.1 Parameter Configuration

Simulation experiments are running on Windows 7, CPU is 3.5GHz Intel Core i7 processor and RAM is 16GB. We write the algorithms with Java, and we use two data

sets, one is a real data set from Board on Geographical Names[1], denoted as GDS, it includes 358957 PoIs. The other one is simulated data set[2], denoted as TDS, this data set is generated by widely used Thomas Brinkhoff Generator which is based on road networks of Oldenburg in Germany, it generates a city area about 24km×27km. The bandwidth between CS and users is 3Mbps. At LSP end, each data set of POIs is indexed by a 2K bytes R-tree structure. The parameter configurations are shown in the following Table 1:

Table 1. Parameters configuration

Parameters	Value range	Defaults
Number of users U	$100000 \le U \le 400000$	300000
Threshold of users at a sensitive place X_T	$100 \le X_T \le 1000$	200
Sensitive places similarity threshold ξ_s	$0 \le \xi_s \le 1$	0.4
Package capacity of PoIs β	$1 \le \beta \le 11$	6
PoIs query number K	$1 \le K \le 15$	8
Distance between user and anchor $dist(q,q')$	$200 \le dist(q,q') \le 1600$	1000

5.2 Success Rate of Anchor Generating

We run the experiments on both data set GDS and TDS, we discuss the success rate of anchor generating when thresholds X_T and ξ_s vary in Formula (4) and Algorithm 2.

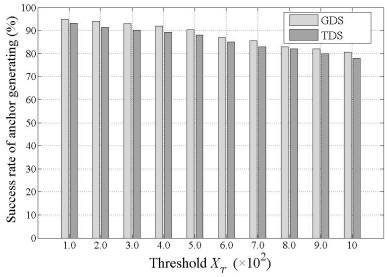

Fig. 7. Threshold X_T varys

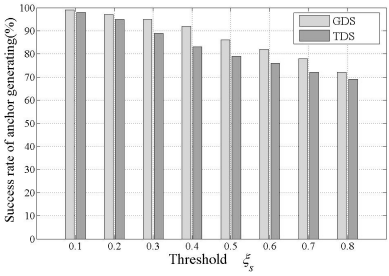

Fig. 8. Threshold ξ_s varys

In Fig.7, when X_T increases, success rate of anchor generating is coming down and keeps stable around 80%, that is due to some places with smaller visiting number are not considered sensitive any more, in a valid region, CS is hard to find enough sensitive places around the user. To the same in Fig. 8, when similarity threshold is increasing, the sensitive places around a user must be similar enough to visiting number and visiting time, it means some places will be filtered. So the anchor generating is affected by these factors.

[1] http://geonames.usgs.gov/index.html
[2] http://iapg.jade-hs.de/personen/brinkhoff/generator/

5.3 Compare with SpaceTwist

We compares our Algorithm 2 to SpaceTwist on data set GDS and TDS, and mainly discuss the communication cost when K and $dist(q,q')$ are changing.

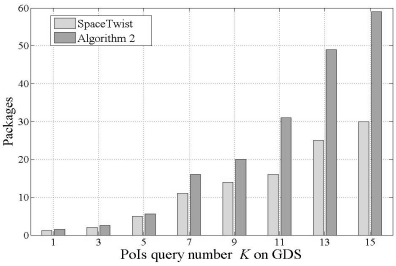

Fig. 9. K varies on GSD

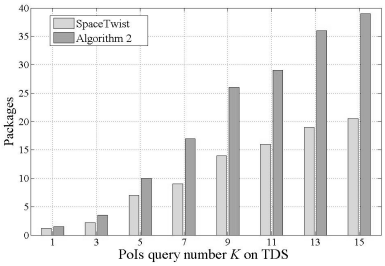

Fig. 10. K varies on TDS

As shown in Fig.9-10, when K is increasing, packages are going up on both data set, and Algorithm 2 is higher than SpaceTwist, especially K varies from 11-15, packages are nearly twice than SpaceTwist. That is due to our algorithm expands demand space and continue query until supply space covers it again. LSP has to continue returning POIs until precise KNN POIs are obtained by CS, therefore the communication is increasing, and when K becomes larger, LSP needs to search more area to get enough POIs, packages are even more. Although Algorithm 2 has higher communication, it is much more precise than SpaceTwist, because the POIs found in our algorithm are around a user rather than the anchor, but SpaceTwist's are all around the anchor q', as shown in Fig.6(b) and Fig.6 (e), our algorithm pays a little more in communication but earns a lot in service quality. Due to demand space expanding, Algorithm 2 can get precise KNN POIs around a user in nearly 100% success rate.

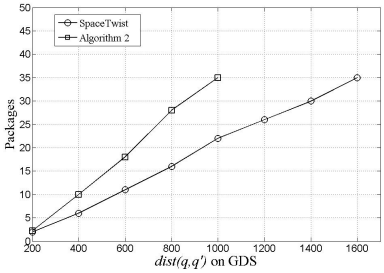

Fig. 11. $dist(q,q')$ varies on GSD

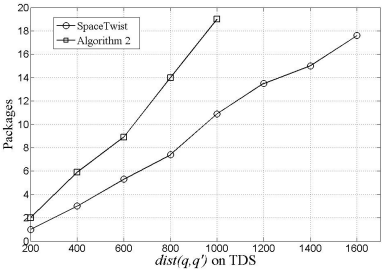

Fig. 12. $dist(q,q')$ varies on TDS

As shown in Fig.11-12, when an anchor is further from the user, LSP has to search a large area to get enough POIs, so its communication increases on both data set, as we discuss the anchor generating in our algorithm is not far away from a user based on grids, that ensures the communication cost of Algorithm 2 is in a reasonable range, in our experiments, we suppose there is no more than 1000 meters between neighbor grids. Communication of Algorithm 2 is higher than SpaceTwist, because it searches a larger area as demand space expands.

6 Conclusions

For location privacy preserving when a user is in a sensitive area, we propose an anchor generating method using a user's neighbor sensitive places to achieve *l*-diversity. By filtering places unsatisfied, CS generates an anchor and uses it to replaces a user's actual location in a query. As the anchor locates at an overlap area of several sensitive places, it increases the probability of appearing at different sensitive places for a user, it avoids the leakage of location privacy when a user and her anchor are both in the same sensitive area. In the query phase, CS needn't submit any user's actual location instead of the generated anchor. According to the POIs set returned by LSP, CS can deduce precise *K*NN POIs around a user, which is much more precise than SpaceTwist. Experiments and performance analysis show that our method is better in security and quality aspects, and its complexity and communication are in a reasonable range.

At the same time we also have some defects such as the factors to define sensitive place are single, we only consider user visiting number and its variation tendency. There is also a defect that the deployment of CS is not discussed, since when a CS is confronting lots of users, the response time may be a bottleneck for the CS. We will focus on these problems in our future works.

Acknowledgements. This research is supported by a grant from National Natural Science Foundation of China (No. 61170241, 61073042), The Fundamental Research Funds for the Central Universities (HEUCFZ1105), Specialized Research Fund for the Doctoral Program of Higher Education (No. 20132304110017), Excellent Youth Foundation of Heilongjiang Province in China (No. JC 201117), Science and Technology Research Project of Heilongjiang Education Department (No. 12513049, NO. 12541788), and this paper is also funded by the International Exchange Program of Harbin Engineering University for Innovation-oriented Talents Cultivation.

References

1. Gruteser, M., Grunwald, D.: Anonymous usage of location-based services through spatial and temporal cloaking. In: Proceedings of the 1st International Conference on Mobile Systems, Applications and Services, pp. 31–42. ACM (2003)
2. Gedik, B., Liu, L.: Location privacy in mobile systems: A personalized anonymization model. In: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems, ICDCS 2005, pp. 620–629. IEEE (2005)
3. Chow, C.Y., Mokbel, M.F.: Trajectory privacy in location-based services and data publication. ACM SIGKDD Explorations Newsletter 13(1), 19–29 (2011)
4. Bamba, B., Liu, L., Pesti, P., et al.: Supporting anonymous location queries in mobile environments with privacygrid. In: Proceedings of the 17th International Conference on World Wide Web, pp. 237–246. ACM (2008)
5. Xu, J., Xu, M., Lin, X., et al.: Location privacy protection through anonymous cells in road network. Journal of Zhejiang University (Engineering Science) 3, 006 (2011)

6. Kido, H., Yanagisawa, Y., Satoh, T.: An anonymous communication technique using dummies for location-based services. In: Proceedings of the International Conference on Pervasive Services, ICPS 2005, pp. 88–97. IEEE (2005)
7. Lu, H., Jensen, C.S., Yiu, M.L.: Pad: Privacy-area aware, dummy-based location privacy in mobile services. In: Proceedings of the Seventh ACM International Workshop on Data Engineering for Wireless and Mobile Access, pp. 16–23. ACM (2008)
8. Hong, J.I., Landay, J.A.: An architecture for privacy-sensitive ubiquitous computing. In: Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services, pp. 177–189. ACM (2004)
9. Yiu, M.L., Jensen, C.S., Huang, X., et al.: Spacetwist: Managing the trade-offs among location privacy, query performance, and query accuracy in mobile services. In: IEEE 24th International Conference on Data Engineering, ICDE 2008, pp. 366–375. IEEE (2008)
10. Pellegrini, S., Ess, A., Schindler, K., et al.: You'll never walk alone: Modeling social behavior for multi-target tracking. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 261–268. IEEE (2009)
11. Huo, Z., Meng, X., Hu, H., Huang, Y.: *y*ou *c*an *w*alk *a*lone: Trajectory privacy-preserving through significant stays protection. In: Lee, S.-g., Peng, Z., Zhou, X., Moon, Y.-S., Unland, R., Yoo, J. (eds.) DASFAA 2012, Part I. LNCS, vol. 7238, pp. 351–366. Springer, Heidelberg (2012)
12. Mokbel, M.F.: Towards privacy-aware location-based database servers. In: Proceedings of the 22nd International Conference on Data Engineering Workshops, pp. 93–93. IEEE (2006)
13. Chow, C.Y., Mokbel, M.F., Liu, X.: A peer-to-peer spatial cloaking algorithm for anonymous location-based service. In: Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems, pp. 171–178. ACM (2006)
14. Chow, C.Y., Mokbel, M.F., Liu, X.: Spatial cloaking for anonymous location-based services in mobile peer-to-peer environments. GeoInformatica 15(2), 351–380 (2011)
15. Huang, Y., Huo, Z., Meng, X.F.: Coprivacy: A collaborative location privacy-preserving method without cloaking region. Jisuanji Xuebao(Chinese Journal of Computers) 34(10), 1976–1985(2011)
16. Gong, Z., Sun, G.Z., Xie, X.: Protecting privacy in location-based services using k-anonymity without cloaked region. In: Mobile 2010 Eleventh International Conference on Data Management (MDM), pp. 366–371. IEEE (2010)
17. Xue, M., Kalnis, P., Pung, H.K.: Location diversity: Enhanced privacy protection in location based services. In: Choudhury, T., Quigley, A., Strang, T., Suginuma, K. (eds.) LoCA 2009. LNCS, vol. 5561, pp. 70–87. Springer, Heidelberg (2009)
18. Xue, J., Liu, X.Y., Yang, X.C., et al.: A location privacy preserving approach on road network. Jisuanji Xuebao(Chinese Journal of Computers) 34(5), 865–878 (2011)
19. Liu, F., Hua, K.A., Cai, Y.: Query l-diversity in location-based services. In: Tenth International Conference on Mobile Data Management: Systems, Services and Middleware, MDM 2009, pp. 436–442. IEEE (2009)