Dummy-Based Trajectory Privacy Protection Against Exposure Location Attacks

Xiangyu Liu[1,2(✉)], Jinmei Chen[1], Xiufeng Xia[1], Chuanyu Zong[1], Rui Zhu[1], and Jiajia Li[1]

[1] School of Computer Science, Shenyang Aerospace University, Shenyang 110136, Liaoning, China
liuxy@sau.edu.cn

[2] School of IT and Business, Wellington Institute of Technology, Lower Hutt 5010, New Zealand
Xiangyu.Liu@weltec.ac.nz

Abstract. With the development of positioning technology and location-aware devices, moving objects' location and trajectory information have been collected and published, resulting in serious personal privacy leakage. Existing dummy trajectory privacy preserving method does not consider user's exposure locations, which causes the adversary can easily exclude the dummy trajectories, resulting in a significant reduction in privacy protection. To solve this problem, we propose a dummy-based trajectory privacy protection scheme, which hides the real trajectory by constructing dummy trajectories, considering the spatio-temporal constraints of geographical environment of the user, the exposure locations in trajectory and the distance between dummy trajectories and real trajectory. We design a number of techniques to improve the performance of the scheme. We have conducted an empirical study to evaluate our algorithms and the results show that our method can effectively protect the user's trajectory privacy with high data utility.

Keywords: Trajectory · Data publishing · Dummy · Privacy protection

1 Introduction

In recent years, with an increasing popularity of positioning technology and location-aware devices, the location and trajectory information of moving objects have been collected and published. Many new applications have emerged because of the mining and analysis of trajectory information. For example, Investors can make business decisions by analyzing trajectory information of users in a specific area, such as where to build a mall. At the same time, government agencies can optimize the design of traffic management systems and traffic routes by analyzing vehicle trajectories in cities. Although publishing of trajectory information plays a significant role in its mobility-related decisions, it also causes serious threats to personal privacy. If a malicious

The work is partially supported by Key Projects of Natural Science Foundation of Liaoning Province (No. 20170520321) and the National Natural Science Foundation of China (Nos. 61502316, 61702344).

W. Ni et al. (Eds.): WISA 2019, LNCS 11817, pp. 368–381, 2019.
https://doi.org/10.1007/978-3-030-30952-7_37

adversary obtains trajectory information, he can get privacy information of user through data mining technology [1–3], such as: home address, hobbies, living habits and health conditions, sensitive relationship etc. Therefore, the privacy protection of trajectory information has been widely concerned by scholars at home and abroad.

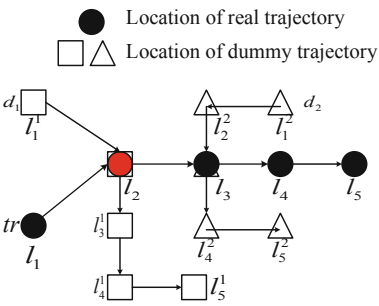
The dummy based trajectory privacy protection method has been widely used in practical research due to its simplicity, no need for trusted third-party entities, and the ability to retain complete trajectory information. However, existing dummy based trajectory protection method does not consider user's exposure location when generating the dummy trajectory, which may reduce the effect of anonymous protection or even directly reveal user's true trajectory. Figure 1(a) shows an example of user's real trajectory $tr = \{(l_1, t_1), (\boldsymbol{l_2}, \boldsymbol{t_2}), (l_3, t_3), (l_4, t_4), (l_5, t_5)\}$, he posts a dynamic through Weibo in location l_2 at time t_2, showing that he is now in location l_2, this information can be obtained by the adversary. When the user exploits existing dummy based trajectory privacy protection scheme to protect his real trajectory, adversary can use this exposure location l_2 to identify some false trajectories. As shown in Fig. 1(b), there are two dummy trajectories generated by algorithm in [14] and a real trajectory. Since the trajectory d_2 does not pass the location l_2 at t_2, adversary can identify it as a false trajectory, so the probability of identifying real trajectory becomes $\frac{1}{2}$, which is greater than anonymous requirement of $\frac{1}{3}$, resulting in user trajectory privacy leakage. We define this attack model as exposure location attack (the specific definition is given later).

id	trajectory
tr	$l_1 \rightarrow l_2 \rightarrow l_3 \rightarrow l_4 \rightarrow l_5$

(a) Real trajectory tr

id	trajectory
tr	$l_1 \rightarrow l_2 \rightarrow l_3 \rightarrow l_4 \rightarrow l_5$
d_1	$l_1^1 \rightarrow l_2 \rightarrow l_3^1 \rightarrow l_4^1 \rightarrow l_5^1$
d_2	$l_1^2 \rightarrow l_2 \rightarrow l_3 \rightarrow l_4^2 \rightarrow l_5^1$

(c) Anonymized trajectory set TRS



(b) Dummy trajectories of tr

Fig. 1. A real trajectory tr, dummy trajectories of tr generated by random algorithm and the anonymized trajectory set TRS of tr

Aiming at this problem, we propose a dummy based trajectory privacy protection scheme. The basic idea is to construct $k-1$ dummy trajectories that are similar to real trajectory and contain exposure locations to hide the real trajectory. It is worth mentioning that the trajectory privacy protection means to protect both whole real trajectory not to be re-identified and sensitive locations (locations of real trajectory except exposure locations) not to be exposed. Figure 1(c) shows the trajectory set after adding dummy trajectories, each dummy trajectory contains the exposure location l_2. The experimental results show that our algorithms can effectively protect the user's real trajectory.

The rest of this paper is organized as follows. Section 2 reviews some related works and Sect. 3 provides preliminaries. Section 4 presents the main steps of DTPP, including generate dummy locations and construct dummy trajectories. Section 5 is devoted to the experimental results. Finally, Sect. 6 concludes this paper.

2 Related Work

Trajectory privacy protection is mainly classified into following categories: dummy trajectories, trajectory *k*-anonymity and trajectory suppression. In [6] Yarovoy et al. proposed two algorithms to generate anonymity groups that satisfies the novel *k*-anonymity. In [14], Moreale et al. proposed a method based on spatial generalization and k-anonymity to transform the original GPS trajectory to achieve anonymity in trajectory dataset. In [7], Abul et al. presented a method named Never walk Alone (*NWA*) to achieve (k, δ)-anonymity through trajectory clustering and space translation. In [8], Huo et al. proposed an approach called You Can Walk Alone *(YCWA*) to protect trajectory privacy through generalization of stay points.

The trajectory suppression method is to selectively publish trajectory data by removing sensitive or frequently accessed locations. In [9], Terrvitis et al. deviced a data suppression technique, which protect privacy while keeping the posted data as accurate as possible. In [10], Zhao et al. proposed two methods based on frequency in trajectories publishing to improve the utility of anonymous data.

The basic idea of dummy trajectory privacy protection was first put forward by Kido et al. in [11, 15]. In [12], Lei et al. proposed two ways to generate false trajectories, namely random pattern and intersection pattern-based scheme. In [13], Lei et al. argued that spatio-temporal correlation should be considered when generating dummy trajectory. In [4], Wu et al. hold that generating dummy trajectory should considering the user mobility pattern and propose a method to protect trajectory privacy based on gravity mobility pattern.

However, the existing dummy trajectory privacy protection methods does not take into account the user's exposure location when generating dummy trajectories, making it easy for the adversary to identify some dummy trajectories or even real trajectory by using the exposure location. Based on this, we design the dummy trajectory privacy protection scheme.

3 Preliminaries

In this section, we define the concepts and notations used throughout the paper. In this paper, a location is a point of interest on the map (e.g. hospital, restaurant, store, bank, etc.), it can be expressed as $l = (x, y)$, where x is the longitude and y is the latitude of location l, we directly use l to represent the location. A trajectory tr is a sequence of n locations, it can be expressed as $tr = \{(l_1, t_1), (l_2, t_2), \ldots (l_n, t_n)\}$, where (l_i, t_i) represents the user checked in the location l_i at t_i, the dummy trajectory is described as $d = \{(l'_1, t_1), (l'_1, t_2), \ldots (l'_n, t_n)\}$. The length of trajectory tr is denoted by $|tr|$. The trajectory set formed by tr and $k-1$ dummy trajectories is defined as $TRS = \{d_1, d_2, \ldots d_{k-1}, tr\}$.

Exposure Location: Given real trajectory $tr = \{(l_1, t_1), \ldots, (\lambda, t_\lambda), \ldots (l_n, t_n)\}$, if the user posts λ to the social network at time t_λ which all people can know it, we say λ is a exposure location of tr.

It is worth mentioning that there may be multiple exposure locations defined in the trajectory, we use $EL = \{(\lambda_1, t_{\lambda_1}), \ldots (\lambda_m, t_{\lambda_m})\}$ (m < n) to represent the set of exposure locations of tr, other locations in trajectory except exposure locations are sensitive locations. As shown in Fig. 1(a), l_2 is the exposure location, and other locations $\{l_1, l_3, l_4, l_5\}$ in the trajectory are sensitive locations.

Exposure Location Attack: Assume *TRS* be trajectory set with respect to tr, and the exposure location set of tr is *EL*. The adversary utilizes the exposure location $(\lambda_i, t_{\lambda_i}) \in EL$ as priori knowledge to attack the trajectory set *TRS*, when a trajectory in *TRS* does not pass the exposure location λ_i at time t_{λ_i}, it can be identified by adversary as a false trajectory, resulting in user trajectory privacy leakage. We define this type of attack as exposure location attack.

For example, as shown in Fig. 1(b), the trajectory d_2 is identified as a false trajectory because it does not pass the location l_2 at time t_2.

Trajectory Leakage Rate (*TE*): Given the trajectory set *TRS* with respect to tr, the adversary uses his background knowledge to identify the false trajectories of *TRS* and the probability of predicting user's real trajectory is defined as follows:

$$TE = \frac{1}{|TRS| - |TRS'|}$$

where $|TRS'|$ indicates the number of false trajectories identified by the adversary.

Average Location Leakage Rate (*LE*): Given the trajectory set *TRS* with respect to tr, the length of tr is n, and the location set containing the real and dummy locations of *TRS* at t_i is L_i, location leakage rate of anyone location in trajectory tr at t_i is defined as $\frac{1}{|L_i|}$, so average location leakage rate is defined as follows:

$$LE = \frac{1}{n} \sum\nolimits_{i=1}^{n} \frac{1}{|L_i|}$$

(p, k)-anonymity: Given the trajectory set *TRS* with respect to tr, and anonymity threshold p, k, if the location leakage rate of anyone location in trajectory tr is not greater than $\frac{1}{p}$, the trajectory leakage rate is not more than $\frac{1}{k}$, we say that the trajectory set *TRS* satisfies (p, k)-anonymity.

For example, as shown in Fig. 1(c), the trajectory set satisfies $(2, 3)$-anonymity.

Location Distance: Given two locations l_i and l_j, the location distance is defined as Euclidean distance between them.

$$dist(l_i, l_j) = \sqrt{(l_i \cdot x - l_j \cdot x)^2 + (l_i \cdot y - l_j \cdot y)^2}$$

Trajectory Distance: Given two trajectories $tr_1 = \{(l_1, t_1), (l_2, t_1), \ldots (l_n, t_n)\}$, $tr_2 = \{(l'_1, t_1), (l'_2, t_2), \ldots (l'_n, t_n)\} (1 \le i \le n)$, the trajectory distance is defined as follows:

$$TDist(tr_1, tr_2) = \sum_{i=1}^{n} dist\left(l_i, l_i^{'}\right) (1 \le i \le n)$$

4 Dummy-Based Trajectory Privacy Protection Scheme

In this section, we present dummy-based trajectory privacy protection algorithm (denoted as DTPP). The main idea of algorithm 1 is to select $k-1$ dummy trajectories to form anonymous trajectory set with real trajectory. In this paper, we hold the view to protect both trajectory and location privacy. Therefore, each dummy trajectory needs to be similar to the real trajectory in shape as much as possible, and can increase the number of dummy locations for sensitive locations in the trajectory set. Algorithm DTPP first obtains a list $Cand_{tr}$ of candidate dummy trajectories sorted by *Score* in descending order (line 2), where *Score* is a heuristic function that measures the impact on both trajectory similarity and location diversity. A dummy trajectory with higher *Score* indicates that more trajectory similarity and location diversity would be achieved by its generation. Then DTPP runs a loop (lines 3–5) while $k-1$ dummy trajectories have been generated, and it attempts to select a dummy trajectory with highest *Score*, which is selected from the top one of $Cand_{tr}$. After getting $k-1$ dummy trajectories, DTPP examines whether the $|L_i|$ of sensitive location satisfies the location anonymity threshold p, if not, it indicates that the dummy locations of this location cannot make the trajectory set satisfy (p,k)-anonymity (lines 7–9). So we suppress the location to ensure the user's location privacy. Finally DTPP returns the anonymous trajectory set (line 10). The details will be introduced in the followings.

Algorithm1: Dummy-based trajectory privacy protection algorithm (DTPP)

Input: Real trajectory tr, exposure location set EL, anonymity threshold (p,k) trajectory distance threshold (α,β),
Output: Trajectory set TRS

1. $TRS \leftarrow \emptyset$;
2. $Cand_{tr}$ = DTC(tr,α,β, EL);//a list of candidate dummy trajectories sorted by *Score* in descending order;
3. **while** $|TRS| \neq k-1$ **do**
4. Select the top one from $Cand_{tr}$ to TRS;
5. Update $Cand_{tr}$;
6. $TRS = TRS \cup tr$;
7. **for** each sensitive location set L_i in TRS **do**
8. **if** $|L_i| < p$ **then**
9. Delete L_i;
10. **return** TRS;

Metric for a Dummy Trajectory. In this section, we consider a goodness metric for a dummy trajectory. We use $d = \{l'_1, l'_2, \ldots l'_n\}$ to represent a dummy trajectory, the effect of a dummy trajectory is summarized by "trajectory similarity", denoted by $Sim(d, tr)$, and the "location diversity", denoted by $Div(d, TRS)$. To maximize the effect of dummy trajectory, we designed a heuristic function as shown in Eq. (1) to select the dummy trajectory.

$$Score = Sim(d, tr) \times Div(d, TRS) \tag{1}$$

Trajectory similarity $Sim(d, tr)$ is defined as Eq. (2), it shows how similar the dummy trajectory is to the real trajectory, and we measure it by the standard deviation of the location distance between the dummy trajectory and real trajectory. In order to match the semantics, we take it countdown. The larger the $Sim(d, tr)$, the more similar dummy trajectory is to the real trajectory.

$$Sim(d, tr) = \frac{1}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(dist(l_i, l'_i) - \frac{1}{n}\sum_{i=1}^{n} dist(l_i, l'_i)\right)^2}} \tag{2}$$

The larger the $|L_i|$ of sensitive location in the TRS, the lower the location leak rate. We use location diversity $Div(d, TRS)$ to measure the effect of dummy trajectory to LE. It is defined as Eq. (3), it indicates the proportion of location sets with increasing amounts after adding the dummy trajectory. Where $div(l'_i, TRS_i)$ is the change of location set at time t_i after adding the dummy location l'_i, there is a change of 1, no change is 0.

$$Div(d, TRS) = \frac{\sum_{i=1}^{n} div(l'_i, TRS_i)}{|tr|} \tag{3}$$

4.1 Generating Dummy Trajectory Candidate Set

In this section, we present Algorithm 2 to generate dummy trajectory candidate set. We propose to construct a dummy trajectory through connecting dummy locations. Suppose a trajectory with s sensitive locations, which generates m dummy locations at each sensitive location, if the enumeration method is used to generate dummy trajectories, there exists m^s trajectories, the number of dummy trajectories increases exponentially with the number of sensitive locations. Considering that the real trajectory has spatio-temporal characteristics, spatio-temporal reachability should be satisfied between adjacent locations of the dummy trajectory. Therefore, we present to model the dummy trajectory candidate set as a directed graph according to whether the adjacent locations of the dummy trajectories is reachable, and formalize it as $G = \{V, E\}$, which will greatly reduce the number of dummy trajectory candidates. V is a set of locations, E is a set of edges. Spatio-temporal reachability is judged by formula (4), where t_i and t_{i+1} represent the timestamps of accessing locations l_i and l_{i+1} respectively, v_{max} is the

user's maximum speed. Obviously, if the formula (4) does not hold, it means that the user could not attend location $v_{i+1,j}$ before t_{i+1} when he starts moving from t_i.

$$\frac{dist(v_{ij}, v_{i+1,j})}{v_{max}} \leq (t_{i+1} - t_i)(1 \leq i \leq n, 0 \leq j \leq |Cand_{l_i}|) \tag{4}$$

As shown in Fig. 2, there is a trajectory directed graph of dummy trajectory candidate set for real trajectory in Fig. 1(a), wherein the location v_{20} is an exposure location. If the adjacent locations are reachable, there is an edge between them. From the directed graph, we can get all possible trajectories.

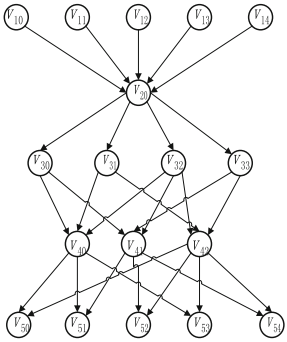


Fig. 2. Trajectory directed graph

Algorithm 2: Generate dummy trajectory candidate set(DTC)

Input: Real trajectory tr, trajectory distance threshold (α,β), exposure location set EL, location anonymity threshold p

Output: Dummy trajectory candidate set $Cand_{tr}$

1. $Cand_{tr} \leftarrow \emptyset$;
2. $Cand_l$ = DLC(tr,β, EL, p);
3. $L_0 = l_0 \cup Cand_{l_0}$;
4. G=TDG(tr,$Cand_l$); // directed graph with spatiotemporal characteristics
5. **for** each location l_i in L_0 **do**
6. Performs a Depth-First Search(DFS) taking l_i as a starting point on the G, and insert trajectories into TR;
7. **for** each trajectory tr_i in TR **do**
8. **if** $\alpha \leq TDist(tr, tr_i) \leq \beta$ **then**
9. Calculate Score=Score(tr_i);
10. Insert <tr_i,Score(tr_i)> into $Cand_{tr}$;
11. **return** $Cand_{tr}$;

In Algorithm 2, each dummy trajectory uses the location l_0 or a dummy location of $Cand_{l_0}$ as starting point and terminates with l_n or a location in $Cand_{l_n}$, DTC performs a Depth-First Search (DFS) taking $l_i \in \{l_0 \cup Cand_{l_0}\}$ as a starting point on the G, then can get all dummy trajectory candidates (lines 5–6). If the distance between dummy trajectory and real trajectory conforms to trajectory distance threshold (α, β), calculates *Score* of the dummy trajectory, and inserts <tr_i, Score (tr_i)> to the dummy trajectory candidate set $Cand_{tr}$ (lines 7–10). Finally, returns $Cand_{tr}$ (line 11).

Algorithm 3: Trajectory directed graph(TDG)

Input: Real trajectory tr, dummy location candidate set $Cand_l$
Output: Trajectory directed graph G

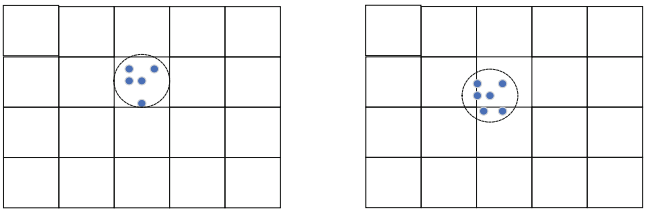
1. $G \leftarrow \emptyset$;
2. v_{max} is the max speed of tr;
3. **for** (i=0; i<$|tr|$; i++) **do**
4. $V_i = l_i \cup Cand_{l_i}$;
5. **for** (i=0; i<$|tr|$-1; i++) **do**
6. **for** each location v_{ij} in V_i **do**
7. **for** each location $v_{i+1,j}$ in V_{i+1} **do**
8. Calculate $dist(v_{ij}, v_{i+1,j})$;
9. **if** $\frac{dist(v_{ij}, v_{i+1,j})}{v_{max}} \leq (t_{i+1} - t_i)$ **then**
10. Insert E<$v_{ij}, v_{i+1,j}$>;
11. $G_{ij} \leftarrow <v_{ij}, E < v_{ij}, v_{i+1,j} >>$;
12. Insert G_{ij} into G_i;
13. Insert G_i into G;
14. **return** G;

The algorithm TDG first merges l_i and $Cand_{l_i}$ into V_i to obtain all points in the directed graph (lines 3–4). Then it iterates all points and judges the spatiotemporal reachability between adjacent locations, if it is satisfied, there exists an edge $E\langle v_{ij}, v_{i+1,j}\rangle$, and finally returns the directed graph G (lines 5–13).

4.2 Generating Dummy Location Candidate Set

In this section, we propose algorithm DLC to generate dummy location candidate set. In the real world, the adversary can obtain the map information from Internet, thus, he can easily exclude the dummy locations according to the geographic feature of the area the dummy locations belong to. For example, if the adversary have captured a location of user is a lake, he can derive that it is a dummy location. So we advocate using real and meaningful locations on the map as dummy locations to protect privacy.

In order to improve the operational efficiency, we propose to divide the map based on grid. The grid increment is set to 2β according to the trajectory distance threshold (α, β). As shown in Fig. 3(a), if the sensitive location is just at the center of the grid, only need to inquire a grid; as shown in Fig. 3(b), if the sensitive location is not in the center of the grid, need to demand four grids, the time is greatly shortened.



(a) sensitive location at the center of grid (b) sensitive location not at the center of grid

Fig. 3. Query grids according to sensitive location

Algorithm 4: Generate dummy location candidate set(DLC)

Input: Real trajectory tr, trajectory distance threshold β, exposure location set EL, location anonymity threshold p

Output: Dummy location candidate set $Cand_l$

```
1.  Cand_l ← ∅;
2.  Divide the map into grids with a size of 2β;
3.  for each location l_i in tr do
4.      if l_i contains in EL then
5.          Cand_{l_i} = ∅;
6.          Insert Cand_{l_i} to Cand_l;
7.      else
8.          Query grids according to coordinate;
9.          for each location l_i' in girds do
10.             if dist(l_i, l_i') ≤ β then
11.                 Insert l_i' to Cand_{l_i};
12.         if |Cand_{l_i}| < p then
13.             delete l_i in tr;
14.         else
15.             Insert Cand_{l_i} to Cand_l;
16. return Cand_l;
```

In algorithm DLC, the whole map of California is uniformly divided into grids with size of 2β (line 2). For each location in the trajectory, if it is an exposure location, its dummy location candidate set is $\varnothing$; if not, query grids according to location coordinate and if the location l_i' of grids satisfies $dist(l_i, l_i') \leq \beta$, add this location to the candidate set $Cand_{l_i}$(lines 3–11). If $|Cand_{l_i}| < p$, it indicates that the region where l_i belongs to is sparse, and cannot generate enough dummy locations to anonymize it. So we suppress it to ensure the user's location privacy (lines 12–13).

5 Experiments

In this section, we provide extensive experiments to evaluate our methods. The user's trajectory data comes from two real datasets:*Gowalla* and *Brightkite*, we also obtain the map data of California, which contains 21,047 nodes and 21,692 edges. In this paper,

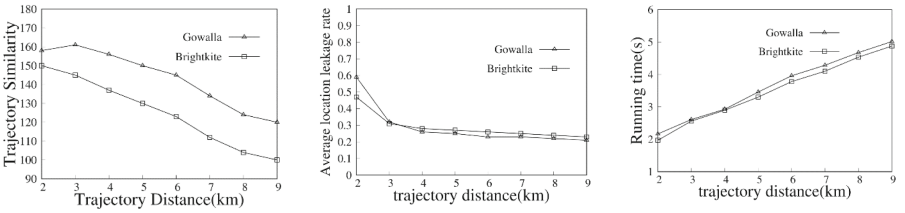
we select 5,000 trajectories of 5,000 users from two datasets respectively for experiment. Table 1 shows the statistics of the experimental data.

We first conduct experiments to obtain the optimal trajectory distance threshold of the algorithm (DTPP), then evaluate the performance of DTPP by comparing with the algorithm in [4], denoted by GM, and the algorithm in [14], denoted by SM. Finally, we evaluate the influence of the parameters only involved in our algorithm. All programs were implemented in Java and performed on a 2.33 GHz Intel Core 2 Duo CPU with 4 GB DRAM running the Windows 7 operating system. The location anonymity threshold p is set $[6, 27]$ (default value 15), the trajectory anonymity threshold k is set $[3, 9]$(default value 3), and the number of exposure location set EL is set $[1, 4]$ (default value 1). We obtain the optimal trajectory distance threshold by testing the trajectory similarity, average location leakage rate (LE) and running time of DTPP at different trajectory distances.

Table 1. Statistics of datasets

	Gowalla	*Birghtkite*
Number of users	5000	5000
Number of trajectories	5000	5000
Number of locations	42683	38916
Average length of trajectory	8.536	7.7832
Total time of trajectories (h)	14734	12176
Total distance of trajectories (km)	40560	37916
Maximum speed of users (km/min)	1.13	1.22

Figure 4 shows that: (1) with the trajectory distance increases, the trajectory similarity and the average location leakage rate gradually decreases, the running time gradually increases. This is because DLC generate more dummy locations when trajectory distance is larger. (2) When the trajectory distance is 2 km, the average location leakage rate is very large although the trajectory similarity is high, the *Gowalla* even reaches 60%; when the trajectory distance is greater than 6 km, the trajectory similarity is reduced, but the average location leakage rate is basically stable at 20%. (3) When the trajectory distance is less than 6 km, the running time of the algorithm is within 4 s. So we set the trajectory distance threshold (α, β) to (3, 6). (4) From the comparison of the two datasets, the *Gowalla* dataset is better than *Birghtkite* in terms of trajectory similarity and average positional leakage rate, but the running time is longer than *Birghtkite*, that is due to the location distribution density in *Gowalla* dataset is slightly higher than *Birghtkite*.

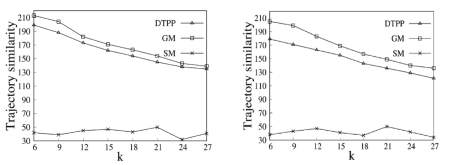


(a)Trajectory similarity (b) Average location leakage rate (c) Running time

Fig. 4. Trajectory similarity, average location leakage rate and running time with varying trajectory distance

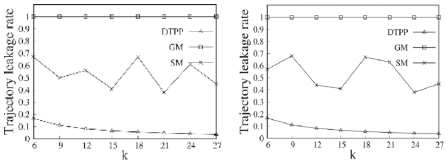
We compare the performance of the algorithms DTPP, GM and SM by three metrics: trajectory leakage rate, trajectory similarity and running time.

It can be seen from Figs. 5, 6 and 7 that: (1) The trajectory similarity of DTPP is about 5% lower than GM, this is because DTPP considers the exposure locations, resulting in a difference in the location distance between the dummy trajectory and the real trajectory larger than GM, but it is about 4 times higher than SM. (2) When there is an exposure location in real trajectory, the adversary cannot identify any false trajectory because DTPP considers this exposure location; while the adversary can uniquely identify the real trajectory in GM due to it does not intersect with the real trajectory when generating dummy trajectories; SM randomly generates a dummy trajectory, so it may contain the exposure location, but the average trajectory leakage rate is as high as about 50%. (3) The running time of DTPP is almost the same as RM, which is lower than 5 s, but much lower than GM. This is because the algorithm GM generates dummy trajectories by using the enumeration method, so it's running time increases exponentially with the number of locations. DTPP greatly reduces the dummy trajectory candidate set by constructing a reasonable data structure, which saves a lot of time. Combining the above points, it can be concluded that the algorithm DTPP can maintain high trajectory similarity and consume little running time while considering the exposure locations, so the algorithm DTPP can effectively protect the user's real trajectory.



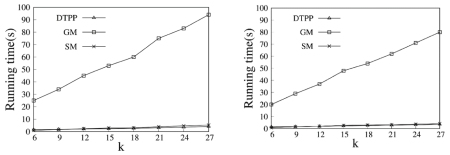
(a) *Gowalla* (b) *Birghtkite*

Fig. 5. Trajectory similarity with varying k



(a) *Gowalla* (b) *Birghtkite*

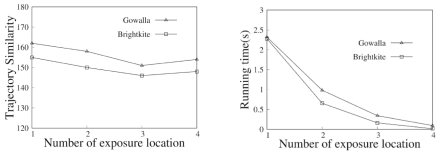
Fig. 6. Trajectory leakage rate with varying k



(a) *Gowalla* (b) *Birghtkite*

Fig. 7. Running time of different protection models

Next, we evaluate the influence of the parameters only involved in our algorithm. First, evaluate the impact of the number of exposure locations on trajectory similarity and running time. Then use the location suppression ratio to measure the impact of the location anonymity threshold p on the algorithm. The location suppression ratio is defined as the percentage of locations that are suppressed.



(a)Trajectory Similarity (b) Running time

Fig. 8. The effect of $|EL|$ on DTPP

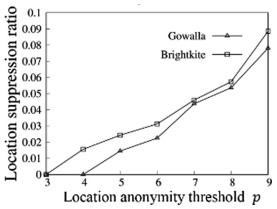


Fig. 9. The effect of p on DTPP

It can be seen from Fig. 8 that: (1) when the number of exposure location $|EL|$ is between 1 and 3, trajectory similarity decreases with increasing the number of exposure location, while $|EL| > 3$, the trajectory similarity increases. This is because when the number of exposure location is increased to a certain extent, the overlapping portions between the dummy and real trajectory become more, so the trajectory similarity is correspondingly improved. (2) The running time of the algorithm DTPP decreases greatly with the increase of the number of exposure location. When $|EL| > 2$, the running time is lower than 2 s, this is because the more the number of exposure location, the fewer dummy locations need to be generated, so the overall time consumption is very small.

As shown in Fig. 9, the location suppression ratio increases with the increase of the location anonymity threshold p, but the location suppression ratio does not exceed 1%, indicating that the algorithm does not need to suppress too many locations to satisfy the location anonymity threshold.

6 Conclusions

In this paper, we propose to consider user's exposure locations in the case of using dummy trajectories to protect trajectory privacy for the first time, and based on this, a trajectory privacy protection algorithm (DTPP) is studied. The algorithm generates the dummy location candidate set based on girds, constructs the trajectory directed graph to store the dummy trajectory candidate set, and establishes a heuristic rule to select the dummy trajectory, which making dummy trajectories have better trajectory similarity while protecting real trajectory. Experiments based on real trajectory datasets show that the algorithm DTPP can effectively protect the trajectory privacy against exposure location attacks.

References

1. Gao, S., Ma, J., Sun, C., Li, X.: Balancing trajectory privacy and data utility using a personalized anonymization model. Netw. Comput. Appl. **38**, 125–134 (2013)
2. Fechner, T., Kray, C.: Attacking location privacy: exploring human strategies. In: ACM Conference on Ubiquitous Computing, pp. 95–98 (2012)
3. Liu, X., Li, M., Xia, X., Li, J., Zong, C., Zhu, R.: Spatio-temporal features based sensitive relationship protection in social networks. In: Meng, X., Li, R., Wang, K., Niu, B., Wang, X., Zhao, G. (eds.) WISA 2018. LNCS, vol. 11242, pp. 330–343. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02934-0_31
4. Wu, Q., Liu, H.X., Zhang, C., Fan, Q., Li, Z.Q., Wang, K.: Trajectory protection schemes based on a gravity mobility model in IoT. Electronics **8**(2), 148–166 (2019)
5. Li, F.H., Zhang, C., Niu, B., Li, H., Hua, J.F., Shi, G.Z.: Efficient scheme for user's trajectory privacy. J. Commun. **36**(12), 114–123 (2015)
6. Yarovoy, R., Bonchi, F., Lakshmanan, S., Wang, W.H.: Anonymizing moving objects: how to hide a MOB in a crowd? In: Kersten, M.L., Novikov, J. (eds.) EDBT 2009, pp. 72–83. ACM Press, New York (2009)
7. Abul, O., Bonchi, F., Nanni, M.: Never walk alone: uncertainty for anonymity in moving objects databases. In: 24th IEEE International Conference on Data Engineering, pp. 215–226. IEEE Press, Washington (2008)
8. Huo, Z., Meng, X., Hu, H., Huang, Y.: You Can Walk Alone: 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. LNCS, vol. 7238, pp. 351–366. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29038-1_26
9. Terrovitis, M., Mamoulis, N.: Privacy preservation in the publication of trajectories. In: 9th International Conference on Mobile Data Management (MDM), pp. 65–72. IEEE (2008)
10. Zhao, J., Zhang, Y., Li, X.H., Ma, J.F.: A trajectory privacy protection approach via trajectory frequency suppression. J. Comput. **37**(10), 2096–2106 (2014)
11. Kido, H., Yanagisawa, Y., Satoh, T.: An anonymous communication technique using dummies for location-based services. In: International Conference on Pervasive Services (ICPS), pp. 88–97. IEEE (2005)
12. Lei, P.R., Peng, W.C., Su, I.J., et al.: Dummy-based schemes for protecting movement trajectories. J. Inf. Sci. Eng. **28**(2), 335–350 (2012)

13. Lei, K.Y., Li, X.H., Liu, H., Pei, Z.X., Ma, J.F., Li, H.: Dummy trajectory privacy protection scheme for trajectory publishing based on the spatiotemporal correlation. J. Commun. **37** (12), 156–164 (2016)
14. Moreale, A., et al.: Movement data anonymity through generalization. IEEE Trans. Data Privacy **3**, 91–121 (2010)
15. Kido, H., Yanagisawa, Y., Satoh, T.: Protection of location privacy using dummies for location-based services. In: 21st International Conference on Data Engineering Workshops, p. 1248. IEEE (2005)