IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 28, 2018, accepted April 19, 2018, date of publication April 25, 2018, date of current version May 16, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2829898

Dummy Generation Based on User-Movement Estimation for Location Privacy Protection

SHUHEI HAYASHIDA[1], DAICHI AMAGATA[1], (Member, IEEE), TAKAHIRO HARA[1], (Senior Member, IEEE), AND XING XIE[2], (Senior Member, IEEE)

[1]Department of Multimedia Engineering, Graduate School of Information Science and Technology, Osaka University, Suita 5650871, Japan
[2]Microsoft Research Asia, Beijing 100080, China

Corresponding author: Daichi Amagata (amagata.daichi@ist.osaka-u.ac.jp)

This work was supported in part by the JSPS Grant-in-Aid for Scientific Research(A) under Grant JP26240013 and in part by the JSPS Grant-in-Aid for Challenging Exploratory Research under Grant JP16K12429.

ABSTRACT Location-based services (LBSs) have been becoming more common due to the prevalence of GPS-enabled devices. While LBSs bring many benefits for our daily lives, location information may reveal private information, rendering an important problem of protecting location privacy of users. To anonymize the locations of users, we focus on dummy-based approaches that generate dummies and their locations are sent along with the actual location of a user to an LBS provider. Although several existing studies developed dummy-based techniques, they assume unrealistic user mobility, e.g., users keep moving and do not stop or follow a pre-defined movement plan precisely. In this paper, we remove the unrealistic assumptions and require much easier input with respect to user movement, i.e., only a set of visiting points. Under the assumption, we propose a dummy generation method, estimation-based dummy trajectory generation (Edge). Based on the given visiting points, Edge estimates a user-movement plan and designs trajectories of dummies so that the adversaries cannot distinguish the user from dummies. We conduct extensive experiments using real map information, and the results show the efficiency and effectiveness of Edge.

INDEX TERMS Location-based services, location privacy.

I. INTRODUCTION

Location based services (LBSs) have been becoming more common due to the prevalence of GPS-enabled devices. Local searches [3], [16], location searches [1], [2], [6], and route planning [33] are the representatives of LBSs. While LBSs bring many benefits for our daily lives, location information may reveal private information [31]. An LBS provider, for example, may be able to identify the addresses of users' homes and their schools/working places, which is actually warned by Krumm [21]. His experiments have confirmed that it is possible to estimate the location of a user's home within a range of 60 meters by only using the last location he/she used that day. The situation is more serious when users continuously utilize LBSs, such as searching nearby points of interests (PoIs) while visiting a city. If LBSs accumulate location histories of users, private locations may be easily mined, and some untrusted LBSs (adversaries) may release private information on users. To avoid this, in LBS usage scenarios, protecting location privacy of users is an important problem. Note that location privacy is defined as the ability to prevent other parties from learning one's current or past location [5].

Numerous approaches so far have been developed to preserve user location privacy. Among them, a representative approach is to generate dummy trajectories [14], [17], [18], [20], [24]–[28], [32], [34]. When a user utilizes a LBS, dummy locations are also sent to the LBS provider. This approach holds the following good properties [15]. (i) Closed system: a dummy-based system is able to be executed on mobile devices held by users, preventing location information leak. (ii) Not disrupting benefits to users: users can receive the service without sacrificing QoS.

Trajectories of dummies have to follow constraints in a real environment so that adversaries who have the map information cannot distinguish the locations of users from those of dummies. For instance, if a dummy moves very fast or is generated at a lake, it is easy to identify that it is a dummy. Although dummy-based approaches have been receiving significant research attention, existing works [20], [25] do not take into account the above important constraint. To address this problem, literatures [14], [17], [18], [32] proposed some dummy-based location anonymization techniques. However, these techniques have a critical drawback. They assume that users keep moving and do not stop [14], [32], and user-movements are precisely known in advance [17], [18]. It is obvious that the above assumptions are hard to hold in practice.

In this paper, we remove the unrealistic assumptions and require much easier input w.r.t. user-movement, i.e., *only a set of visiting points*. Under the assumption, we propose a dummy generation method, Edge (Estimation-based dummy trajectory generation). To protect location privacy of users, it is important to (i) avoid the situation that adversaries can trace the user-trajectory and (ii) distribute dummies widely to make the user-existing area uncertain. However, designing dummies that can satisfy the above requirements is challenging, since we do not know the actual (future) user-movement when we generate dummies. Edge overcomes this challenge based on the following approaches. Edge firstly estimates the user's visiting order of the given visiting points (user-movement plan) with the traveling salesman problem. Here, we can consider an approach that generates dummy trajectories based on the whole user-movement plan (the estimated user trajectory from the first visiting point to the last one). This however may not make sense, since the estimated user-movement plan may be incorrect. To address this problem and efficiently adapt to the actual user-movement, every time the user arrives at a visiting point, Edge estimates candidates of the next visiting point and designs trajectories of dummies. For each candidate of the next visiting point, trajectories of some dummies are designed so that they intersect with the estimated user-trajectory. The others are designed so that the user and the dummies exist in a wide area, by a greedy approach.

A. CONTRIBUTIONS

The contribution of this paper is twofold. (i) We remove the unrealistic assumptions of existing studies and require only a set of visiting points w.r.t. user-movement. We propose Edge, which generates trajectories of dummies under this situation. Edge can anonymize the user-positions effectively, as well as reduce the burden of a user because it is not necessary to input the details of his/her movement plan. (ii) We conduct simulation experiments using real map information to validate the effectiveness of Edge. The experimental results show that Edge outperforms the existing work.

B. ROAD-MAP

The rest of this paper is organized as follows. In Section II, we review related work, and we provide preliminary information for designing our method in Section III. Section IV describes Edge. We conduct simulation experiments in Section V, and Section VI concludes this paper.

II. RELATED WORK

There are numerous studies on protecting location privacy. We can categorize them into four approaches: (i) k-anonymity, (ii) Mix Zone, (iii) Obfuscation, and (iv) Dummy generation.

A. k-ANONYMITY

This approach guarantees that the location of a user is mixed with at least k candidates, i.e., the location of a user cannot be identified over the probability of $\frac{1}{k}$ [9], [12], [13], [22], [23]. This approach collects the locations of k users and sends the minimum region including these k users to an LBS server as a query instead of the exact location of the user. This approach basically needs to pool the locations of users, thus the above literatures assume a trusted third-party server to mediate interactions between the users and the LBS server [13], or utilize peer-to-peer collaboration between mobile users [9]. It is however difficult in practice to deploy a completely safe third-party server. In addition, mobile peer-to-peer collaboration suffers from the same problem of location privacy, since users have to share their location information with others they do not know. Another main drawback of this approach is that anonymizing the location of a user results in failure if the number of users around him/her is insufficient.

B. MIX ZONE

Literatures [5], [11], [29] employing Mix Zone assume that users utilize LBSs consecutively in a short time. In such an environment, adversaries may be able to track the user-trajectory by connecting the locations submitted by the user. That is, if the location of the user is identified once, the adversaries can easily track the user by the above approach. A Mix Zone is an area in which users cannot utilize LBSs. Given users existing in a Mix Zone, the third-party server exchanges identifiers (or names) of the users so that the adversaries cannot trace users, even if the locations of the users have been identified at a certain timing. If the size of Mix Zone is large, location privacy of users can be easily protected. However, users cannot utilize LBSs in a long term, if they exist in a large Mix Zone, which degrades QoS. The third-party server in addition has to be trusted, thus this approach is also difficult to be deployed in practice.

C. OBFUSCATION

This approach replaces the location of a user with a nearby intersection or building to obscure his/her real location [4], [7], [10]. However, if there are no appropriate targets around the user, the substitute location is far from that of the user, which also degrades QoS. Literature [19] employs Hilbert curves to transform the location of a user and sends the transformed location to the LBS provider. The transformation is one-way, thereby the LBS provider cannot decode the location of the user. The disadvantage of this approach is that it also needs LBS providers to transform all their location data (such as locations of shops), which is not a trivial effort in maintaining services.

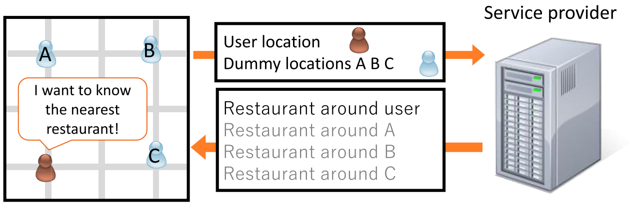


FIGURE 1. Example of dummy-based approach.

D. DUMMY GENERATION

This section introduces existing dummy-based approaches. As described in Section I, this approach generates dummies and sends their locations along with the actual location of a user to an LBS server. For example, Fig. 1 illustrates that a user sends its current location with some dummy-locations to retrieve the nearest restaurant while protecting its location privacy. The LBS provider then returns a list of restaurants that are the nearest to each location of the user and dummies. The user finds the nearest restaurant from the list by ordering the results based on the distances from his/her location.

Literatures [14], [32] generate dummies around the user in a grid pattern while considering physical constraints in a real environment. However, this technique holds an unrealistic assumption that users keep moving without stops. Literatures [17], [18] assume that users move while stopping at some locations, and generates dummies to anonymize user-positions. In this method, users are required to input their movement plans (e.g., trajectory, pause-time, pause-position) before using LBSs, and precisely follow the plan. In other words, the user-movements are known in advance. Trajectories of dummies are designed so that they are widely distributed periodically while considering intersections with the user. The assumption described above (i.e. user-movements are precisely known in advance) is however unrealistic because the actual user-movements are usually different from the registered plan. For example, a user changes the order of visiting points and/or walks more slowly due to the crowdedness.

Edge does not require the unrealistic assumption and requires much easier input, which is an advantage over the existing methods. Furthermore, our experimental results show that Edge outperforms the existing work [17], although we do not know the exact user-movement plan. Some works also propose dummy generation methods, but they consider either shapshot case (i.e., non-consecutive queries) [31] or ad-hoc measurement [24], [26], which are different from our work.

III. PRELIMINARY

In this section, we first describe the assumption w.r.t. LBSs in Section III-A, and present the requirements to protect location privacy in Section III-B.

A. ASSUMPTION

1) LBS USAGE

Users hold devices (e.g., smartphones) that have installed map information. We assume that users utilize LBSs and service requests are sent consecutively. Users register a set of their visiting points before utilizing LBSs. When a user utilizes LBSs, his/her device sends his/her location information along with the locations of dummies.

2) MOBILITY MODEL

Users move and stop at some locations. More specifically, they walk to some visiting points through the shortest route or a route approaching the destination at random speed of $[v_{min}, v_{max}]$ and stop for a random time of $[t_{min}, t_{max}]$. Users are sure to visit each of all the registered points once. A hybrid mobility model, e.g., users walk and take buses and trains, is out of scope of this paper and remains as a future work. Note that the initial trajectories of dummies are generated when users have arrived at the first visiting points.

3) ADVERSARY MODEL

Adversaries aim at tracking a particular user to obtain his/her private information (e.g., home address). That is, the adversaries try mining the trajectory of the user and eliminating dummies. In this paper, we consider that LBS providers are untrusted thus may be adversaries. Besides, adversaries may know the mobility model and map information.

B. REQUIREMENTS TO PROTECT LOCATION PRIVACY

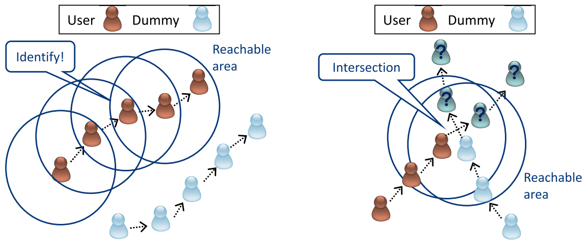
For location privacy protection, we consider the following requirements: (i) *consistency of movement*, (ii) *traceability*, and (iii) *anonymous area*.

1) CONSISTENCY OF MOVEMENT

It is trivial that when generating a location of a dummy, the location has to be reachable from the previous location. Besides, we should consider the actual road networks when calculating the distance between two locations, rather than the Euclidean distance. We also need to exclude areas where people normally do not inhabit, such as seas and forests, as locations for dummies. For example, it is obviously a dummy if it is moving from a pedestrian sidewalk to the center of a highway, although the moving distance is acceptable in terms of its moving speed. We determine locations of dummies by taking into account the actual map information to satisfy these conditions.

2) TRACEABILITY

We should take the traceability of user locations into account when the user utilizes LBSs consecutively in a short period. The trajectories of a user may be estimated from the location history that has been accumulated at an LBS provider. In this paper, traceability means the ability to identify the trajectory of a user by combining consecutive locations during a certain period. The traceability problem becomes particularly serious when the location of the user is accidentally identified, e.g., the user involuntarily uploaded a photo showing where he/she is to a social networking service. If the locations of the user are traceable, all the previous (and possibly future) locations also become obvious. The trajectory of the user is, for



(a) (b)

FIGURE 2. Traceability. (a) An adversary can trace the user because his/her reachable area includes only him/her. (b) Traceability decreases when the trajectory of a dummy intersects with that of the user.

example, easily distinguished from those of dummies when the locations of the user are traceable, as Fig. 2a illustrates. This is because all the reachable areas include only the user. A simple but effective solution to lower the traceability is to design trajectories of dummies so as to intersect with the trajectory of the user [34]. Fig. 2b illustrates such a situation. In Fig. 2b, the locations of a user and dummy are in the same reachable area, thereby it is difficult for adversaries to trace the trajectory of the user.

Here, each queried location i has a probability of being the user location ρ_i. If the user is accidentally identified (suppose its location is j), $\rho_j = 1$. Now we are given two locations x and y, whose probabilities of being the user location are respectively ρ_x and ρ_y. Assume two entities (a user or dummy) exist at x and y, respectively. Assume further that their next positions (locations) are in the same reachable area, then their locations have the $\frac{\rho_x+\rho_y}{2}$ user location probability. Based on this idea, we define mean time to confusion (MTC) according to [30].

Definition 1 (Mean Time to Confusion): Every time a service request is issued, we calculate the entropy of the probability of it being the location of the user by

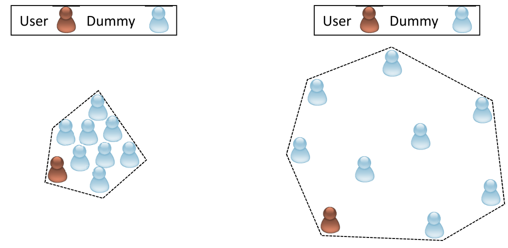
$$H = -\sum_{i \in D} \rho_i \log \rho_i,$$

where ρ_i is the probability of location i being the location of the user and D is the set of all locations corresponding to the user and all dummies. We assume that the LBS provider would sometimes leak the location of the user in our evaluation. Mean time to confusion (MTC) is the mean time period from the time when H becomes zero (when the location of the user is revealed) to the time when H exceeds one (when we can regard the location of the user as being anonymized).

To summarize, MTC is the mean time that is necessary to anonymize the location of a user from an accidental disclosure by an LBS provider, and smaller MTC means lower user traceability.

3) ANONYMOUS AREA

Based on [25], we define the anonymous area to measure the anonymity of the location of a user as a criterion to evaluate how secure a user is.



(a) (b)

FIGURE 3. Anonymous area. (a) Low anonymity. (b) High anonymity.

Definition 2 (Anonymous Area): Given the current locations of a user and dummies, the anonymous area is the convex hull of the set of the locations.

Fig. 3b has better location anonymity than Fig. 3a, i.e., the anonymous area in Fig. 3b is larger than that in Fig. 3a. It is intuitively known that the appropriate size for an anonymous area depends on the situation. We hence allow users (or applications) to specify the size of the anonymous area, and location anonymization techniques attempt to satisfy the specified size.

IV. EDGE

A. OVERVIEW

Assuming a continuous use of an LBS in a short period, it is important that the trajectory of a user is not traceable. Therefore, Edge first generates trajectories of dummies to decrease MTC, and then generates trajectories of dummies for enlarging the anonymous area. Under the situation that the input w.r.t. the user-movement plan is only a set of visiting points, the user moving position is unknown. Edge hence estimates the candidates of user moving positions. Specifically, by solving the traveling salesman problem, we obtain a list of visiting orders of the user's visiting points.[1] (Assume that the user inputs n visiting points.) For example, each entry in the list is described as $vp_1 \rightarrow \cdots \rightarrow vp_{n-1} \rightarrow vp_n$, meaning that the user travels the visiting points in the order of $vp_1, \ldots, vp_{n-1}, vp_n$. Note that each entry has its traveling distance, and the list is sorted in ascending order of the travel distance. Informally, Edge picks α entries with the shortest distance. Assume that the user has just arrived at the ith visiting point, and let $\langle vp_{i+1}^{\alpha'}, vp_{i+2}^{\alpha'} \rangle$ be an ordered pair of the $(i+1)$th and the $(i+2)$th visiting points of the $\alpha'(\leq \alpha)$th entry. Edge firstly generates trajectories of α dummies so that they visit $vp_{i+1}^1, \ldots, vp_{i+1}^{\alpha'}, \ldots,$ and vp_{i+1}^{α}, respectively. Its intuitive idea is that, as long as the user goes toward one of the α points, one of the α dummies will intersect with the user. Recall Definition 1, and to decrease MTC, short intersection interval is important. Edge hence generates additional α dummies so that they visit $vp_{i+2}^1, \ldots, vp_{i+2}^{\alpha'}, \ldots,$ and vp_{i+2}^{α}, respectively. That is, Edge firstly generates 2α dummies to (i) lower the traceability (decrease MTC) and

[1]Edge is orthogonal to any prediction models. If we know the user-movement model, Edge can simply employ it to predict the next visiting points.

(ii) deal with some possible (and promising) visiting orders. Note that users can specify the parameter α. If α is large, the estimation accuracy would be high but communication cost increases (the number of dummies increases). If α is small, on the other hand, the estimation accuracy may not be high. However, our empirical study shows that even when α is small, e.g., $\alpha = 2$, Edge outperforms the existing work which knows the exact user-movement plan.

Besides, Edge further generates β dummies (β is also an parameter that users can specify). Its objective is to enlarge the anonymous area and eliminate a hint for identifying the user. Assume that a method generates 2α dummies based on the above approach and β dummies only for wide anonymous area. In this method, an entity, which frequently intersects with the other entities,[2] is probably the user, which may become a hint. To avoid this, in Edge, trajectories of some of β dummies are generated so that they intersect with other dummies.

Based on the above approaches, m dummy trajectories are generated every time the user has arrived at a visiting point ($m = 2\alpha + \beta$). Since Edge incrementally generates trajectories of dummies, it is probable to adapt the actual user-movement flexibly.

1) I/O OF EDGE

The input is represented by $\{\alpha, \beta, r, VP\}$. To easily specify anonymous area size, users can assume that the area is square. Then, r is the length of one side of the square, i.e., the area size is r^2. $VP = \{vp_1, vp_2, \cdots, vp_{|VP|}\}$ is a set of visiting points vp_i of a user. The output is a set of trajectories of m dummies. Let d_i be the ith dummy. The trajectory of d_i is a set S_i of tuples $\langle p, t\rangle$, where p is a location (e.g., a visiting point or an intersection point) and t is the time when the dummy arrives at p. Given two sequential tuples $\langle p_a, t_a\rangle$ and $\langle p_b, t_b\rangle$, the dummy moves from p_a to p_b at a random speed $\in [v_{min}, v_{max}]$. In addition, if the dummy stays p, its staying time is randomly chosen from $[t_{min}, t_{max}]$.

2) FRAMEWORK

Algorithm 1 shows the framework of Edge. The user inputs α, β, r, and VP, and starts using an LBS. After the user specified the inputs, based on VP, Edge obtains a list L of visiting orders by solving the traveling salesman problem, where the starting points are the first visiting point (line 2). Note that each element (visiting order) in L is sorted in ascending order of the total travel distance.

Every time user has arrived at a visiting point vp_i, Edge estimates α candidates of the next visiting positions and movement plans (line 4). It then designs the trajectories of α dummies to intersect with the user (line 5). Next, according to the number of intersections with the entities (line 6), trajectories of λ dummies are generated for intersection with other dummies (line 7). Also, trajectories of the other dummies are generated to enlarge the anonymous area size (line 8).

[2]This holds if one can exactly count the intersection.

Algorithm 1 Edge

```
   Input: α, β, r, VP
1  case User has specified the parameters
2      L ← Traveling-salesman(VP)
3  case User has arrived at vp_i
4      Est-user-movement-plan(α, L)
5      Gen-dummy-intersection-with-user(α)
6      λ ← Dummy-assignment-calculation(β)
7      Gen-dummy-intersection-with-dummy(S′, λ)
       //S′ is a set of trajectories of dummies for
       intersection with other dummies
8      Gen-dummy-area-enlargement(S_e) //S is a set
       of trajectories of dummies for anonymous area
       enlargement
9  case User sends its location information to the LBS
   provider
10     Check difference between the actual and the
       estimated user movement plans
11     Adjust the arrival time of dummies for intersection
       with the user
```

Thereafter, to intersect with the user and the dummy certainly, we need to adjust the arrival time to the intersection point with the user of the dummy. To this end, we judge whether the estimated user-movement plan is different from the actual one or not. If different, we re-estimate the arrival time. According to this, the trajectory of the dummy is updated (line 11). We elaborate each operation in the following subsections.

B. ESTIMATION OF USER-MOVEMENT PLAN

When the user has arrived at a visiting point vp, Edge estimates α next visiting points and movement plans from L. Below, we describe how to estimate the visiting orders (corresponding to the α next points) and each element in the user-movement plans (i.e., moving route, pause-time, and moving speed).

1) VISITING ORDER

Assume that vp is the actual ith visiting point of the user. Firstly, Edge deletes all entries (visiting orders), from L, whose ith visiting point is not vp. Edge then picks the first α entries so that their $(i+1)$th visiting points are distinct. (In other words, if the $(i+1)$th visiting point of a given entity is the same as ones with higher ranks, the entry is ignored.) As a result, Edge estimates that the $(i+1)$th points in the entries are the next visiting points.

For example, Table 1 illustrate an example of L where $VP = \{A, B, C, D, E, F, G\}$ and the first visiting point is A. Assume that the user has just arrived B and B is the second visiting point. In this case, the 4th and the last entries are removed from L, and Edge focuses only on the 1st, 2nd, 3rd, and 5th entries (omitted entries are also assumed to be

TABLE 1. An example of L where where $VP = \{A, B, C, D, E, F, G\}$ and the first visiting point is A.

Visiting order	Travel distance
$A \to B \to D \to C \to G \to E \to F$	10
$A \to B \to D \to C \to G \to F \to E$	12
$A \to B \to C \to E \to F \to G \to D$	16
$A \to D \to C \to E \to G \to F \to B$	22
$A \to B \to D \to C \to G \to E \to F$	28
$\vdots$	$\vdots$
$A \to F \to G \to C \to B \to D \to E$	51

removed). Assume further that $\alpha = 2$. Since the 2nd entry has the same 3rd visiting point as the 1st entry (i.e., D), the 2nd entry is ignored. Consequently, Edge estimates that the next visiting point is D (in the 1st entry) or C (in the 3rd entry).

2) MOVING ROUTE

We estimate that the moving route is the shortest path between two sequential visiting points.

3) PAUSE-TIME

The pause time at each vp is assumed to be a uniformly distributed random time in $[t_{min}, t_{max}]$.

4) MOVING SPEED

The moving speed between visiting points is also assumed to be a uniformly distributed random time in $[v_{min}, v_{max}]$.

C. GENERATING TRAJECTORIES OF DUMMIES

We describe how to generate trajectories of dummies towards each objective, i.e., (i) intersection with the user, (ii) intersection between dummies, and (iii) anonymous area enlargement. Recall that, for (i), Edge generates trajectories of 2α dummies, and trajectories of β dummies are generated for (ii) and (iii).

1) INTERSECTION WITH USER

As described in Section IV-A, by solving the traveling salesman problem, we have α visiting orders. Assume that the user has just arrived at the ith visiting point. Edge generates trajectories of 2α dummies so that they intersect with the user at the $(i+1)$th and the $(i+2)$th points in the α orders. (That is, if the next visiting point of the user is one of the $(i+1)$th points in the α orders, a dummy will intersect with the user.)

Algorithm 2 illustrates how to generate the trajectories of dummies to intersect with the user. Let $\mathcal{S}$ be the set of m trajectories. We sort each trajectory in $\mathcal{S}$ in ascending order of the number of intersections with the user (line 2). Note that, during the user-traveling, for each dummy, we count the number of intersections with the user. Let $\langle p_{i'}^{\alpha'}, t_{i'}^{\alpha'} \rangle$ be a tuple of the estimated i'th vp and arrival time of the user in the α'th visiting order. In the sorting order, we scan the trajectories of dummies (line 6), and let $\langle p, t \rangle$ be the last generated tuple

Algorithm 2 Gen-Dummy-Intersection-With-User(α)

Input: α
1 $\mathcal{S} \leftarrow \{S_1, S_2, \cdots, S_m\}$
2 Sort $\mathcal{S}$ in ascending order of the number of intersections with the user
3 **for** $i' = i+1$ **to** $i+2$ **do**
4   **for** $\alpha' = 1$ **to** α **do**
5     $\langle p_{i'}^{\alpha'}, t_{i'}^{\alpha'} \rangle \leftarrow$ a tuple of the estimated i'th vp and arrival time of the user in the α'th entry of the visiting order
6   **for** $\forall S_j \in \mathcal{S}$ **do**
7       $\langle p, t \rangle \leftarrow$ a tuple of the last generated position and arrival time in S_j
8   **if** ***Reachable***$(\langle p_{i'}^{\alpha'}, t_{i'}^{\alpha'} \rangle, \langle p, t \rangle)$ **then**
9   **Trajectory-design**$(S_j, \langle p_{i'}^{\alpha'}, t_{i'}^{\alpha'} \rangle)$
10         $\mathcal{S} \leftarrow \mathcal{S} \backslash \{S_j\}$
11   **break**

in S_j. If $p_{i'}^{\alpha'}$ is reachable from p while taking into account $t_{i'}^{\alpha'}$ and t (line 8), we design the trajectory of d_j from p to $p_{i'}^{\alpha'}$ (line 9). Note that we consider that $p_{i'}^{\alpha'}$ is reachable from p, based on the following definition.

2) DEFINITION 3 (REACHABLE AREA)

If $\langle p_i, t_i \rangle$ and $\langle p_j, t_j \rangle$ are within the reachable area, they satisfy

$$dist(p_i, p_j) \leq v_{max} \cdot (t_j - t_i), \ t_i < t_j,$$

where $dist(p_i, p_j)$ is the distance between p_i and p_j on the map.

Furthermore, Trajectory-design$(S_j, \langle p', t' \rangle)$ is a function that generates d_j's trajectory from p to p', where p is the last generated point. In a nutshell, this function considers the shortest path between p and p', and finds some intersection points and PoIs on the path. Let p_i be a location on the path and t_i be a time when d_j arrives at p_i. If p_i is reachable from p, $\langle p_i, t_i \rangle$ is added to S_j. (The moving speed between p and p_i and the time staying p_i are decided accordingly.) By repeating this operation, d_j's trajectory is generated.

3) HOW TO DECIDE λ

Recall that Edge generates trajectories of λ dummies for intersection between dummies and those of $(\beta - \lambda)$ dummies for enlarging anonymous area (see Algorithm 1). Now we explain how to decide λ for each time when the user arrives at a visiting point. Assume that Edge will generate trajectories of d_1–d_β (i.e., the trajectories of $d_{\beta+1}$–d_m have been generated for intersection with the user). The idea here is that to make it hard to identify which entity is a dummy, dummies should evenly intersect with the entities. Based on this, d_1–d_β are sorted in the ascending order of the number of intersections with the entities. We use dummies within $[1, \lfloor \frac{m}{2} \rfloor]$ ranks for intersection with dummies, and use the other dummies for enlarging anonymous area.

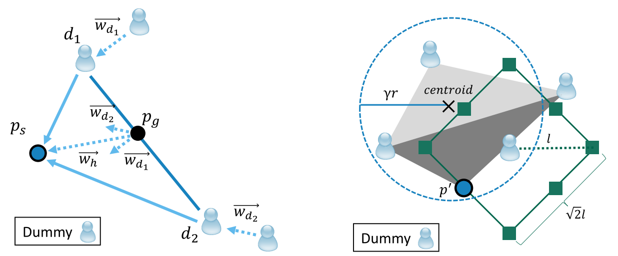
Algorithm 3 Gen-Dummy-Intersection-With-Dummy ($\mathcal{S}'$, λ)

Input: $\mathcal{S}'$ //$\mathcal{S}'$ ($|\mathcal{S}'| = \lambda$) is a set of trajectories of dummies for intersection with other dummies.

1 **for** $\forall S_i \in \mathcal{S}'$ **do**
2 $\quad \langle p, t\rangle \leftarrow$ a tuple of the last generated position and arrival time of d_i
3 $\quad$ **for** $\forall S_j \in \mathcal{S}' \backslash S_i$ **do**
4 $\quad\quad \langle p', t'\rangle \leftarrow$ a tuple of the last generated position and arrival time of d_j
5 $\quad\quad p_g \leftarrow$ a random point on the segment between p and p'
6 $\quad\quad \zeta \leftarrow$ a random value $\in [0, 1]$
7 $\quad\quad p^* \leftarrow \mathrm{argmax}_{p_m \in P} dist(p, p_m) + dist(p', p_m)$ where p_m is a point which is movable both from p and p'
8 $\quad\quad \overrightarrow{w_h} \leftarrow \frac{\zeta \cdot min(dist(p^*, p), dist(p^*, p'))}{|\overrightarrow{w_{d_i}} + \overrightarrow{w_{d_j}}|}(\overrightarrow{w_{d_1}} + \overrightarrow{w_{d_2}})$ //$\overrightarrow{w_{d_i}}$ ($\overrightarrow{w_{d_j}}$) is a vector consisting of moving direction of d_i (d_j) starting at p_g
9 $\quad\quad p_s \leftarrow$ the nearest PoI of $\overrightarrow{w_h}$
10 $\quad\quad t_s \leftarrow$ the estimated arrival time to the next vp of user
11 $\quad\quad$ **if** ***Reachable***$(\langle p, t\rangle, \langle p_s, t_s\rangle) \wedge$ ***Reachable***$(\langle p', t'\rangle, \langle p_s, t_s\rangle)$ **then**
12 $\quad\quad\quad$ **Trajectory-design**$(S_i, \langle p_s, t_s\rangle)$
13 $\quad\quad\quad$ **Trajectory-design**$(S_j, \langle p_s, t_s\rangle)$
14 $\quad\quad\quad \mathcal{S}' \leftarrow \mathcal{S}' \backslash \{S_i, S_j\}$
15 $\quad\quad\quad$ **break**
16 $\quad$ **if** $|\mathcal{S}'| = 1$ **then**
17 $\quad\quad$ Let $\mathcal{S}_e$ be the set of trajectories of dummies for anonymous area enlargement
18 $\quad\quad \mathcal{S}_e \leftarrow \mathcal{S}_e \cup \mathcal{S}'$

4) INTERSECTION BETWEEN DUMMIES

If dummies intersect only with the user, it becomes easier to identify that they are dummies. To avoid this, Edge generates trajectories of λ dummies for intersecting with other dummies, which is described in Algorithm 3. In short, Algorithm 3 finds a pair of dummies d_i and d_j which can share a position. Let p and p' be the last generated points of d_i and d_j, respectively (lines 2 and 4). Now we have a segment between p and p', and we randomly pick a point on the segment, denoted by p_g. We next make a vector $\overrightarrow{w_h}$ starting at p_g, and its direction is $\overrightarrow{w_{d_i}} + \overrightarrow{w_{d_j}}$ (line 8). Also, its size is obtained based on $|\overrightarrow{w_{d_i}}|$, $|\overrightarrow{w_{d_j}}|$, a random variable $\in [0, 1]$, and the common moving area of d_i and d_j. The detail is described at lines 6–8, and an intuitive example is illustrated in Fig. 4a. After that, we obtain the nearest PoI of $\overrightarrow{w_h}$, p_s. If both d_i and d_j are reachable to p_s, Edge designs trajectories of d_i and d_j from p and p' to p_s (lines 12–13). The above operation is repeated until no more pair is generated.



(a) (b)

FIGURE 4. How to decide dummy-moving position. (a) Intersection with another dummy. (b) How to decide dummy-moving position to enlarge the anonymous area.

5) ANONYMOUS AREA ENLARGEMENT

Lastly, Edge generates trajectories of dummies for enlarging the anonymous area. Our approach greedily enlarges the anonymous area to satisfy r^2 size. Algorithm 4 shows its detail. Let $\mathcal{S}_e$ be the set of trajectories of dummies for anonymous area enlargement. In addition, let p be the last generated position of $S \in \mathcal{S}_e$. Given S, Edge calculates the size of the anonymous area consisting of the last generated positions of dummies in $\mathcal{S} \backslash \mathcal{S}_e$, and vp_i (recall that vp_i is the current user-location). If the size is less than r^2, Edge considers a position of d so that the size becomes r^2 as much as possible. (Otherwise, the trajectory of d is generated so that d is not far from the dummies in $\mathcal{S} \backslash \mathcal{S}_e$, see lines 15–17.) Edge computes a rhomboid centered at p and its length is $\sqrt{2}l$, where l is the maximum distance w.r.t. the movable area from p. Fig. 4b illustrates an example. Edge considers eight points, i.e., the vertices of the rhomboid and the mid-points of its edges, as shown in Fig. 4b. Then, Edge calculates the anonymous area size by considering the case where d is at one of the eight points (line 9). Edge selects the point p^*, from the eight points, such that the anonymous area size is close to r^2 the most. Let p' be the nearest PoI of p^*, and the trajectory of d from p to p' is generated (line 13).

Note that p^* satisfies $dist(p^*, p_c) \leq \gamma \cdot r$, where p_c is the centroid of the set of the last generated positions of dummies in $\mathcal{S} \backslash \mathcal{S}_e$ and γ is a constant. Without this constraint, p^* may be very far from the other dummies, which can be a hint to identify that it is a dummy. Besides, due to this constraint, dummies are not distributed wastefully, thereby Edge can design dummies so that both small MTC and appropriate anonymous area size hold.

D. ARRIVAL TIME ADJUSTMENT

Because Edge *estimates* each user-movement plan, it may be incorrect. Thus, whenever the user issues a service request (i.e., sends the current location information), Edge compares each estimated user-movement and the actual one. If they are different, Edge re-estimates the user-movement and adjusts the time when the dummy d arrives at vp. In other words, the trajectory of d, which plans to intersect with the user at vp, is updated so that they intersect with each other. We describe the detail below.

Algorithm 4 Gen-Dummy-Area-Enlargement($\mathcal{S}_e$)

Input: $\mathcal{S}_e$ ($\mathcal{S}_e$ is the set of trajectories of dummies for anonymous area enlargement)

```
for ∀S ∈ S_e do
    P ← a list of the last generated positions of dummies in this term and vp_i
    p ← the last generated position of S
    h ← the size of the convex hull obtained from P ∪ {p}
    t' ← the estimated time when the user arrives at the next vp
    l ← the maximum distance w.r.t. movable area from p
    if h < r^2 then
        Generate a rhomboid, whose length is √2 l, centred at p
        ãĂĂ C ← {⟨p_1, h_1⟩, ··· , ⟨p_8, h_8⟩} //p_j is a vertex or the mid-point on an edge of the rhomboid and h_j is the size of the convex hull obtained from P ∪ {p_j}
        p_c ← the centroid of P
        p* ← argmin_{p_j∈C, γ·r≤dist(p_j,p_c)} |r^2 − h_j|
        p' ← the nearest PoI of p*
        Trajectory-design(S, ⟨p', t'⟩)
    else
        p_n ← the nearest position of p in P
        p' ← a PoI existing in close to the segment between p_n and p
        Trajectory-design(S, ⟨p', t'⟩)
```

1) HOW TO JUDGE THE DIFFERENCE

As mentioned above, based on the actual user-movement, the difference from each estimated one is obtained. The representative differences are moving route, pause-time, and moving speed. Below, we present how to re-estimate the user-movement for each case.

- Moving route. When the user is located at a different route from the estimated location, the time to take to the visiting point vp will become different. Edge therefore re-estimates the time and assumes that the user moves to vp through the shortest path. Based on this, the arrival time is estimated.
- Pause-time. Assume that the user is now moving (or stays at a visiting point) in the estimated plan. When the user sends its location information, it turns out to be that he/she is staying the visiting point (or moving). In this case, we consider that the pause-time increased (or decreased), then estimate that the time when he/she arrives at each visiting point is delayed (or becomes earlier).
- Moving speed. Consider a case that the user is located at the estimated path but the actual location is different from the estimated one. In this case, Edge calculates the moving speed based on the previous and the current positions. Then Edge assumes that the user moves at the speed, and estimates the time when the user arrives at the next visiting point accordingly.

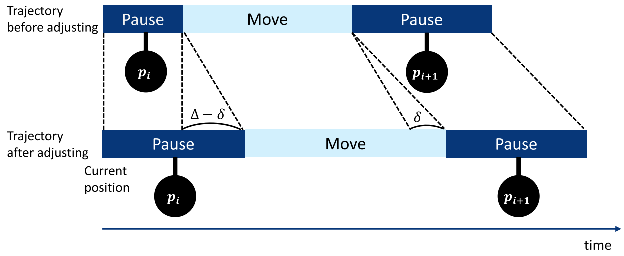


FIGURE 5. Adjustment of the time when the dummy arrives at p_{i+1}.

2) HOW TO UPDATE THE DUMMY TRAJECTORY

If there is a difference as described above, Edge updates the trajectory of the dummy d which plans to intersect with the user. This is important to decrease MTC. Let Δ be the difference w.r.t. the previously estimated time when the user arrives at the next visiting point vp and the newly estimated time. Edge updates the part of the trajectory of d, i.e., $\langle vp, t\rangle \in S$. Specifically, $\langle vp, t\rangle \in S$ is updated to $\langle vp, t + \Delta\rangle$. To this end, w.r.t. d, the pause-time at each point, the moving speed between sequential points, and the moving route are updated.

Fig. 5 illustrates an example. Assume that the dummy, which plans to intersect with the user, stays p_i and the next point is p_{i+1}. Given $\Delta > 0$, Edge adjusts the time when d arrives at p_{i+1} by varying the pause-time at p_i. If this variation δ is less than Δ (e.g., as in Fig. 5), Edge adjusts the moving speed so that the arrival time delays for Δ as much as possible. If this is not enough, Edge searches another moving route to p_{i+1} so that the total variation becomes (or is close to) Δ. Note that if the user is moving, Edge adjusts moving speed of the dummy, and the same approaches are executed.

E. DISCUSSION

In this paper, we assume that adversaries have no background information, although some existing studies assume that querying probabilities of users in a given area [28] and visiting orders based on PoI semantics [8] are known. This is because as long as users use Edge, such background information is hardly obtained (due to entity intersections). If we assume that adversaries know such information, it is possible to extend Edge so that dummies visit plausible PoIs by taking into account query probabilities in a given area and/or PoI semantics. However, optimizing the extension is beyond the scope of this paper, and is considered to be a future work.

V. EXPERIMENTS

To investigate the performance of Edge, we conducted simulation experiments. The objective of experiments is to evaluate the robustness of Edge from the quantitative perspective.

TABLE 2. Configuration of parameters (bolds are default).

Parameter	Range
r[m]	1,000, 1,250, **1,500**, 1,750, 2,000
α	**2**, 3, 4
β	3, 4, **5**, 6, 7
v_{min}[m/sec]	**1.05**
v_{max}[m/sec]	**1.55**
t_{min}[sec]	**600**
t_{max}[sec]	**1,800**

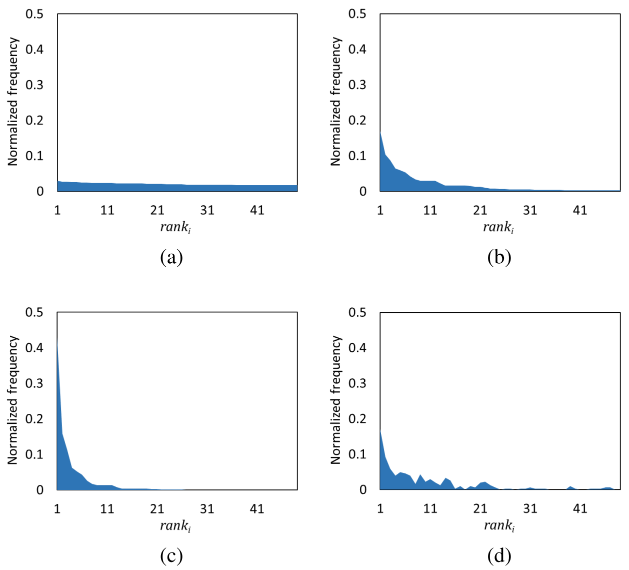


FIGURE 6. User behavior models. (a) Synthetic-low. (b) Synthetic-mid. (c) Synthetic-high. (d) Real.

A. SETTING

We conducted experiments using a map of Tokyo 23 wards (map area: 1,689.6 [km^2], number of visiting points: 2,729,587, number of intersections: 325,946). We allocated the Venue using the Foursquare API[3] to the visiting points, and reproduced the distribution of the actual visiting points on the map. The parameters used in our experiments are described in Table 2, and the default values are described in bold.

We generated three synthetic datasets, Synthetic-low, Synthetic-mid, and Synthetic-high to investigate the impact of an accuracy of user-movement plan. Besides, a real dataset, denoted by Real, is used. The real dataset is the Foursqueare check-in dataset [35]. Given VP, there are $(|VP| - 1)!$ visiting orders, and we rank each order based on the travel distance. Fig. 6 illustrates how often users follow the ith rank order (normalized frequency) for each dataset. For example, in Synthetic-low, users often do not follow good visiting orders w.r.t. travel distance (Fig. 6a), while in Synthetic-high, users usually do (Fig. 6c). We see that, in Synthetic-low (Synthetic-high), the estimation accuracy of next visiting points in Edge may be low (high). In addition, Real is similar to Synthetic-mid. Note that $|VP|$ is fixed to be 7, and for each user trajectory model, we used 50 trajectories.

B. CRITERIA

We used MTC and Anonymous area achieving Ratio-Size (AR-Size) as evaluation criteria. They respectively measure a satisfaction rate of a traceability and a requested anonymous area.

1) MTC

MTC is the value defined by Definition 1. Note that we do not lower the probability of a location being the user's if dummies (and the user) encountered one another from opposite directions on a road and approximately go straight. As with [17], we employ 30° to judge whether we hold this case or not.[4] That is, if the intersection angle of two entities is less than or equal to 30°, we do not allocate user probabilities to the entities.

2) AR-SIZE (ANONYMOUS AREA ACHIEVING RATIO-SIZE)

This is defined as follows.

Definition 4 (Anonymous Area Achieving Ratio-Size) Whenever the user requests an LBS, we calculate the anonymous area size at that time. Let h and c be the sum of the calculated size and the total number of service requests of the user, respectively. Anonymous area achieving Ratio-Size (AR-Size in short) is $\frac{h}{c}$.

If AR-Size is larger than 1, we can regard that a given method is able to obscure the entities' positions more than the user's request on average.

It is worth noting that, even when $m = 15$ ($\alpha = 4$ and $\beta = 7$), the average computation time for generating trajectories of dummies is 27.59 [sec].[5] This suggests that the computation time is small enough in practice (since users normally stay a visiting point longer than the time). Thus we focus on the above two criteria.

C. EVALUATED METHOD

In this experiment, we evaluated the performances of the following three methods.

- **Dum-P-Cycle [17].** This method generates trajectories of dummies based on the assumption that the user-movement plan is precisely known in advance. (The trajectory of each dummy is generated by considering the user trajectory which passes through all the specified visiting points.) To make this method function, we provide this method with the exact user-movement plan, i.e., this method cannot be present in practice, just for reference.
- **E-Dum-P-Cycle** (Estimation-based Dum-P-Cycle)**.** This method has the same assumption as Edge,

[3] https://developer.foursquare.com

[4] The angle for the judging is dependent on applications and situations, thus addressing how to specify an appropriate value is beyond the scope of this paper.

[5] All experiments were conducted on a PC with 3.2GHz Intel Xeon E5-2667 v4 processor.

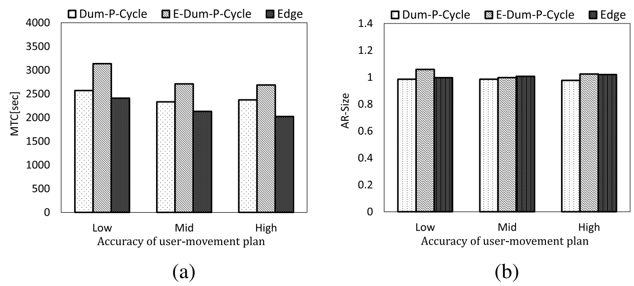


FIGURE 7. MTC and AR-Size vs. Accuracy of user-movement plan. (a) MTC. (b) AR-Size.

i.e., the input w.r.t. user-movement is only *VP*. This method estimates that the user-movement plan is the optimal solution of the traveling salesman problem, and generates trajectories of dummies by employing Dum-P-Cycle. To be fair, E-Dum-P-Cycle also employs the same trajectory update approach as Edge (described in Section IV-D).

- **Edge.** The method proposed in this paper.

Note that each method generates the same number of dummies.

D. RESULT

1) IMPACT OF THE ACCURACY OF USER-MOVEMENT PLAN

First, to investigate the impact of estimation accuracy, we conducted an experiment using the three synthetic datasets, and Fig. 7a shows it result. (Recall that Dum-P-Cycle knows the exact trajectory in advance, i.e., it does not use estimation, and is employed for comparison here.) From Fig. 6a, we see that the MTC of E-Dum-P-Cycle and Edge decreases when the accuracy increases. This is because each entity (user and dummies) can easily intersect with the target ones (i.e., their distance can be small) when the accuracy is high. It is important to note that the MTC of Edge is small even in the case of Synthetic-low. Edge estimates multiple points as the next visiting points, which enables entity intersections effectively. We see from Fig. 7b that the estimation accuracy affects the AR-Size of each method little. Each method distributes dummies so that the anonymous area size is satisfied, which is independent on user-movement, there by this result is obtained.

We next show the results of experiments performed by varying r, α, and β. (We fixed γ as 1.1. Note that we also investigated the impact of γ, and we confirmed that as γ increases, the MTC and AR-Size of Edge increases.) We observed that the tendency of each method on Synthetic-mid and Synthetic-high is similar to that on Real, which can be intuitively seen from Figs. 6 and 7. We therefore omit the results on Synthetic-mid and Synthetic-high to keep this paper concise.

2) VARYING r

We first investigate the impact of r, and Fig. 8 shows the result. The first observation is that the MTC of Edge is the smallest among the three methods, which is shown in Figs. 8a–8b. Recall that when the user has arrived at the ith visiting point, Edge generates trajectories of dummies so that they intersect with the user at the $(i+1)$th and the $(i+2)$th visiting points. The other methods do not consider such a short interval of intersections, although it is important to decrease MTC. Furthermore, Edge outperforms Dum-P-Cycle even in Synthetic, low estimation accuracy model, as shown in Fig. 8b, though Dum-P-Cycle knows the exact user-movement plan. Edge generates 2α dummies for intersection with the user, and this effectively functions for lowering traceability. Besides, the result of E-Dum-P-Cycle shows that generating the *whole trajectories* of dummies based on estimation is not effective. Edge is superior to E-Dum-P-Cycle, since Edge generates trajectories of dummies incrementally to adapt to the actual user-movement. The second observation is that as r increases, the MTC of all the methods increases. Large r means that the user requires large anonymous area. To satisfy this, dummies are distributed widely, so the distance between each entity tends to be long. In this case, to intersect with the user, it takes longer time, resulting in longer MTC.

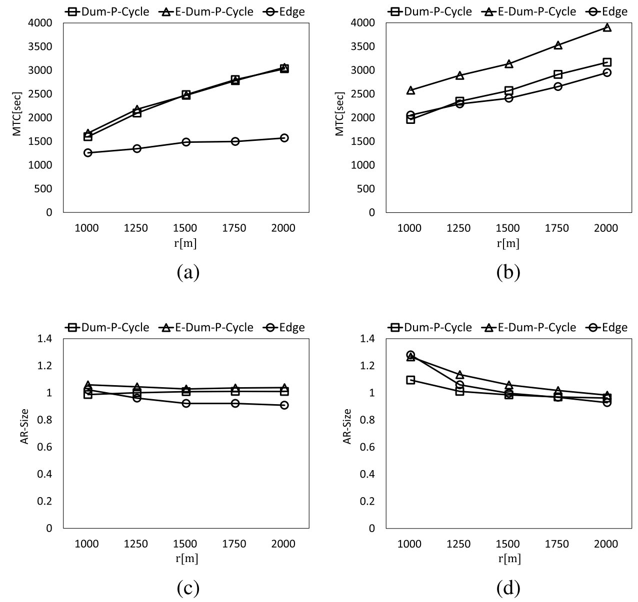


FIGURE 8. MTC and AR-Size vs. r. (a) MTC (Real). (b) MTC (Synthetic-low). (c) AR-Size (Real). (d) AR-Size (Synthetic-low).

Figs. 8c–8d show the AR-Size for each r. We can see that the AR-Size of Edge is approximately 1, meaning that Edge (mostly) satisfies the user-requirement. Note that larger AR-Size does not necessarily mean better performance (if AR-Size is not less than 1), since the user-requirement is r^2 sized anonymous area. That is, the results shown in Figs. 8c–8d do not suggest that Dum-P-Cycle and E-Dum-P-Cycle outperform Edge.

3) VARYING α

We next study the impact of α. Fig. 9 shows the result. As α increases, MTC of all the methods decreases,

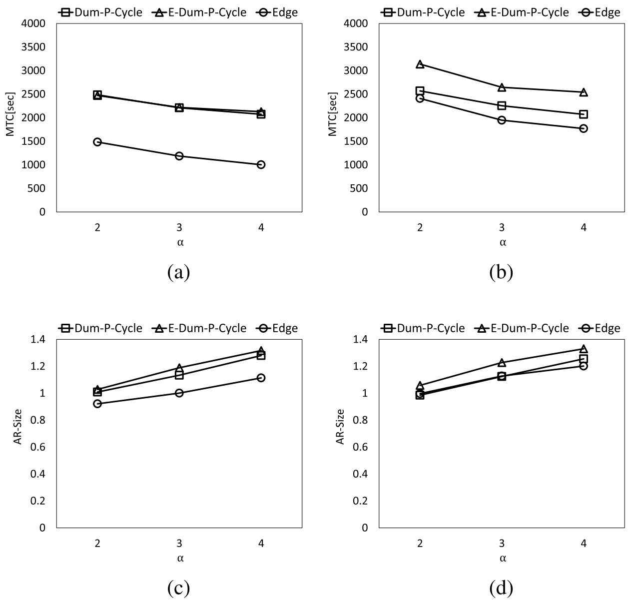


FIGURE 9. MTC and AR-Size vs. α. (a) MTC (Real). (b) MTC (Synthetic-low). (c) AR-Size (Real). (d) AR-Size (Synthetic-low).

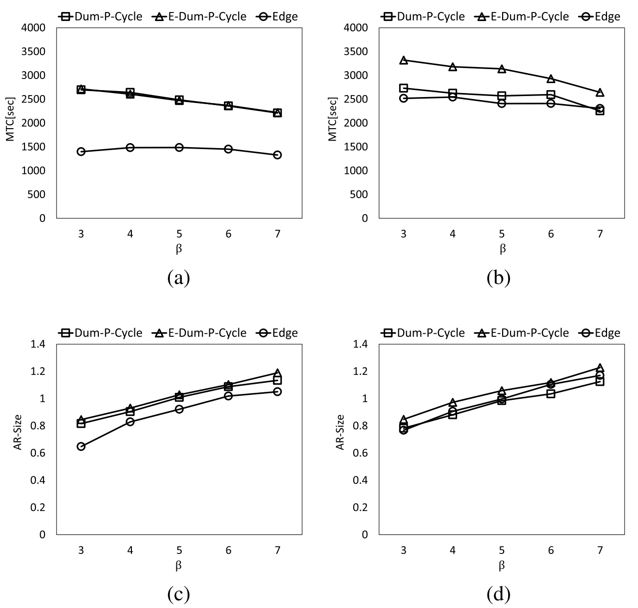


FIGURE 10. MTC and AR-Size vs. β. (a) MTC (Real). (b) MTC (Synthetic-low). (c) AR-Size (Real). (d) AR-Size (Synthetic-low).

as shown in Figs. 9a–9b. This is reasonable, since as α increases, the number of dummies also increases, thus the probability of intersecting with the user becomes larger. We can see from Fig. 9b that large α particularly functions in Synthetic, and Edge effectively lowers the traceability.

Figs. 9c–9d show the result of AR-Size. The AR-Size of all the methods increases as α increases. This is because, as the number of dummies increases, the entities tend to be distributed over a wide area.

4) VARYING β

We finally investigate the impact of β, and the result is shown in Fig. 10. Recall that in Edge, as β increases, the number of dummies for intersecting between dummies and enlarging anonymous area increases. Therefore, as illustrated in Figs. 10a–10b, β does not affect the MTC of Edge so much. It is important to note that Edge keeps outperforming the other methods even in the case where the number of dummies is large.

As noted, β affects the anonymous area size, and Figs. 10c–10d confirm this. The results are similar to those in Figs. 9c–9d, since the tendency appearing in the result is derived from the number of dummies.

VI. CONCLUSION

We studied the important problem of location anonymization in LBS usages and focused on a dummy based approach. Existing studies have unrealistic assumptions, i.e., users keep moving without stops [14], [32] and user-movement plan is known in advance and the user precisely follows the plan [17], [18]. In this paper, we removed the unrealistic assumptions, and users need to input only a set of visiting points w.r.t. their movement. We proposed Edge that can lower the traceability and keep the user-required anonymous area size under the practical assumption. Edge utilizes the traveling salesman problem to estimate the user-movement and designs trajectories of dummies so that they intersect with the user effectively and enlarge the anonymous area size appropriately.

We conducted simulation experiments using real map information. The experimental results show that Edge can lower the traceability more compared with the existing methods. Also, if the number of dummies is sufficient, Edge can satisfy the required anonymous area size.

In this paper, we studied the performances of Edge w.r.t. MTC and AR-Size. It is also important to investigate whether the dummies generated by Edge can be identified by visual observation or not. We therefore plan to conduct a user experiment to evaluate the robustness of Edge against humans.

ACKNOWLEDGMENT

The map data of Tokyo 23 wards used in the experiments are provided by the Japan Digital Road Map Association.

REFERENCES

[1] D. Amagata and T. Hara, "Monitoring maxRS in spatial data streams," in *Proc. EDBT*, 2016, pp. 317–328.
[2] D. Amagata and T. Hara, "A general framework for maxRS and maxCRS monitoring in spatial data streams,"*ACM Trans. Spatial Algorithms Syst.*, vol. 3, no. 1, pp. 1–34, 2017.
[3] D. Amagata, T. Hara, and S. Nishio, "Distributed top-k query processing on multi-dimensional data with keywords," in *Proc. SSDBM*, 2015, pp. 10:1–10:12.
[4] C. A. Ardagna, M. Cremonini, E. Damiani, S. De Capitani di Vimercati, and P. Samarati, "Location privacy protection through obfuscation-based techniques," in *Proc. DBSec*, 2007, pp. 47–60.
[5] A. R. Beresford and F. Stajano, "Location privacy in pervasive computing," *IEEE Pervasive Comput.*, vol. 21 no. 1, pp. 46–55, Jan. 2003.

[6] V. Bindschaedler and R. Shokri, "Synthesizing plausible privacy-preserving location traces," in *Proc. SP*, May 2016, pp. 546–563.
[7] N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Optimal geo-indistinguishable mechanisms for location privacy," in *Proc. CCS*, 2014, pp. 251–262.
[8] S. Chen and H. Shen, "Semantic-aware dummy selection for location privacy preservation," in *Proc. ISPA*, 2016, pp. 752–759.
[9] C.-Y. Chow, M. F. Mokbel, and X. Liu, "A peer-to-peer spatial cloaking algorithm for anonymous location-based service," in *Proc. GIS*, 2006, pp. 171–178.
[10] M. Duckham and L. Kulik, "A formal model of obfuscation and negotiation for location privacy," in *Proc. PervCom*, 2005, pp. 152–170.
[11] J. Freudiger, R. Shokri, and J.-P. Hubaux, "On the optimal placement of mix zones," in *Proc. PETS*, 2009, pp. 216–234.
[12] S. Gao, J. Ma, C. Sun, and X. Li, "Balancing trajectory privacy and data utility using a personalized anonymization model," *J. Netw. Comput. Appl.*, vol. 38, pp. 125–134, Feb. 2014.
[13] B. Gedik and L. Liu, "Location privacy in mobile systems: A personalized anonymization model," in *Proc. ICDCS*, 2005, pp. 620–629.
[14] T. Hara, A. Suzuki, M. Iwata, Y. Arase, and X. Xie, "Dummy-based user location anonymization under real-world constraints," *IEEE Access*, vol. 4, pp. 673–687, 2016.
[15] C. S. Jensen, H. Lu, and M. L. Yiu, "Location privacy techniques in client-server architectures," *Privacy Location-Based Appl.*, 2009, pp. 31–58.
[16] Y. Jing, L. Hu, W.-S. Ku, and C. Shahabi, "Authentication of k nearest neighbor query on road networks," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 6, pp. 1494–1506, Jun. 2014.
[17] R. Kato, M. Iwata, T. Hara, Y. Arase, X. Xie, and S. Nishio, "User location anonymization method for wide deistribution of dummies," in *Proc. DEXA*, 2013, pp. 259–273.
[18] R. Kato *et al.*, "A dummy-based anonymization method based on user trajectory with pauses," in *Proc. GIS*, 2012, pp. 249–258.
[19] A. Khoshgozaran and C. Shahabi, "Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy," in *Proc. SSTD*, 2007, pp. 239–257.
[20] H. Kido, Y. Yanagisawa, and T. Satoh, "An anonymous communication technique using dummies for location-based services," in *Proc. ICPS*, 2005, pp. 88–97.
[21] J. Krumm, "Realistic driving trips for location privacy," in *Proc. PerCom*, 2009, pp. 25–41.
[22] B. Lee, J. Oh, H. Yu, and J. Kim, "Protecting location privacy using location semantics," in *Proc. KDD*, 2011, pp. 1289–1297.
[23] C. Li and B. Palanisamy, "ReverseCloak: Protecting multi-level location privacy over road networks," in *Proc. CIKM*, 2015, pp. 673–682.
[24] H. Liu, X. Li, H. Li, J. Ma, and X. Ma, "Spatiotemporal correlation-aware dummy-based privacy protection scheme for location-based services," in *Proc. INFOCOM*, 2017, pp. 1–9.
[25] H. Lu, C. S. Jensen, and M. L. Yiu, "PAD: Privacy-area aware, dummy-based location privacy in mobile services," in *Proc. MobiDE*, 2008, pp. 16–23.
[26] B. Niu, S. Gao, F. Li, H. Li, and Z. Lu, "Protection of location privacy in continuous LBSs against adversaries with background information," in *Proc. ICNC*, 2016, pp. 1–6.
[27] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Achieving k-anonymity in privacy-aware location-based services," in *Proc. INFOCOM*, 2014, pp. 754–762.
[28] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Enhancing privacy through caching in location-based services," in *Proc. INFOCOM*, 2015, pp. 1017–1025.
[29] B. Palanisamy and L. Liu, "MobiMix: Protecting location privacy with mix-zones over road networks," in *Proc. ICDE*, 2011, pp. 494–505.
[30] R. Shokri, J. Freudiger, M. Jadliwala, and J.-P. Hubaux, "A distortion-based metric for location privacy," in *Proc. WPES*, 2009, pp. 21–30.
[31] R. Shokri, G. Theodorakopoulos, G. Danezis, J.-P. Hubaux, and J.-Y. Le Boudec, "Quantifying location privacy: The case of sporadic location exposure," in *Proc. PETS*, 2011, pp. 57–76.
[32] A. Suzuki, M. Iwata, Y. Arase, T. Hara, X. Xie, and S. Nishio, "A user location anonymization method for location based services in a real environment," in *Proc. GIS*, 2010, pp. 398–401.
[33] C. R. Vicente, I. Assent, and C. S. Jensen, "Effective privacy-preserving online route planning," in *Proc. MDM*, 2011, pp. 119–128.
[34] Y. Yanagisawa, H. Kido, and T. Satoht, "Location privacy of users in location-based services," *Mobiquitous*, 2006, pp. 1–4.
[35] D. Yang, D. Zhang, V. W. Zheng, and Z. Yu, "Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs," in *Proc. IEEE SMC*, vol. 45, no. 1, pp. 129–142, Jan. 2015.

SHUHEI HAYASHIDA received the B.E. and M.Sc. degrees from Osaka University, Japan, in 2016 and 2018, respectively. His research interests include location privacy.

DAICHI AMAGATA (M'15) received the B.E, M.Sc., and Ph.D. degrees from Osaka University, Osaka, Japan, in 2012, 2014, and 2015, respectively. He is currently an Assistant Professor with the Department of Multimedia Engineering, Graduate School of Information Science and Technology, Osaka University. His research interests include distributed and parallel query processing, data monitoring over stream environments, and mobile computing. He is a member of ACM.

TAKAHIRO HARA (SM'98) received the B.E., M.E., and Dr.Eng. degrees in information systems engineering from Osaka University, Osaka, Japan, in 1995, 1997, and 2000, respectively. He is currently a Full Professor with the Department of Multimedia Engineering, Osaka University. His research interests include distributed databases, peer-to-peer systems, mobile networks, and mobile computing systems. He is a member of three other learned societies. He is a Distinguished Scientist of ACM.

XING XIE (SM'04) received the B.S. and Ph.D. degrees in computer science from the University of Science and Technology of China, in 1996 and 2001, respectively. He joined Microsoft Research Asia in 2001, where he was involved in spatial data mining, location-based services, social networks, and ubiquitous computing. He is currently a Senior Research Manager with Microsoft Research Asia and a Guest Ph.D. Adviser with the University of Science and Technology of China. He is a Senior Member of ACM and a Distinguished Member of the China Computer Federation.