Where to Meet a Driver Privately: Recommending Pick-Up Locations for Ride-Hailing Services

Chen, Yifei ; Li, Meng; Zheng, Shuli ; Lal, C.; Conti, M.

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Where to Meet a Driver Privately: Recommending Pick-Up Locations for Ride-Hailing Services

Yifei Chen[1,2,3,4], Meng Li[1,2,3,4]($\boxtimes$), Shuli Zheng[1,2,3,4], Chhagan Lal[5], and Mauro Conti[6]

[1] Key Laboratory of Knowledge Engineering with Big Data (Hefei University of Technology), Ministry of Education, Hefei, China
`yifeichen@mail.hfut.edu.cn, mengli@hfut.edu.cn`
[2] School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China
[3] Anhui Province Key Laboratory of Industry Safety and Emergency Technology, Hefei University of Technology, Hefei, China
[4] Intelligent Interconnected Systems Laboratory of Anhui Province, Hefei University of Technology, Hefei, China
[5] Delft University of Technology, Delft, Netherlands
`c.lal@tudelft.nl`
[6] Department of Mathematics, University of Padua, 35131 Padua, Italy
`conti@math.unipd.it`

Abstract. Ride-Hailing Service (RHS) has motivated the rise of innovative transportation services. It enables riders to hail a cab or private vehicle at the roadside by sending a ride request to the Ride-Hailing Service Provider (RHSP). Such a request collects rider's real-time locations, which incur serious privacy concerns for riders. While there are many location privacy-preserving mechanisms in the literature, few of them consider *mobility patterns* or *location semantics* in RHS. In this work, we propose a pick-up location recommendation scheme with location indistinguishability and semantic indistinguishability for RHS. Specifically, we give formal definitions of location indistinguishability and semantic indistinguishability. We model the rider mobility as a time-dependent first-order Markov chain and generates a rider's mobility profile. Next, it calculates the geographic similarity between riders by using the Mallows distance and classifies them into different geographic groups. To comprehend the semantics of a location, it extracts such information through user-generated content from two popular social networks and obtains the semantic representations of locations. Cosine similarity and unified hypergraph are used to compute the semantic similarities between locations. Finally, it outputs a set of recommended pick-up locations. To evaluate the performance, we build our mobility model over the real-world dataset GeoLife, analyze the computational costs of a rider, show the utility, and implement it on an Android smartphone. The experimental results show that it costs less than 0.12 ms to recommend 10 pick-up locations within 500 m of walking distance.

1 Introduction

Ride-hailing service [19, 20, 25] (RHS) is now a ubiquitous application in vehicular networks [22, 23, 31]. It enables riders to be matched with available drivers in their vicinity [10]. A rider meets a driver at a pick-up location and they drive toward the rider's destination. To complete the matching between riders and drivers, a Ride-Hailing Service Provider (RHSP) is required and successful RHSPs include Uber and Didi. According to a report from Statista, RHSs enable 78 million people to enjoy rides using the Uber app on a monthly basis [24].

To find a driver, the rider has to upload a pick-up location to the RHSP for notifying the drivers in the area covering this location. However, location is highly related to rider's sensitive locations, e.g., home and work, and it calls for proper sanitation before sharing it with the RHSP. Furthermore, there are attacks against riders' location privacy, such as location inference attack [27] and membership inclusion attack [8].

Among all the ride activities, riders tend to hail a ride from the same location frequently as shown in Fig. 1. For example, Alice takes a cab to work every morning on weekdays. This observation is also supported by two latest works which call it spatiotemporal activity [9] and similar query [18]. There has been a large body of work on designing a Location Privacy-Preserving Mechanism (LPPM) [21]. In this work, we aim to protect riders' location privacy in such a setting, i.e., we protect the *true location* that could be masked by possible pick-up locations recommended via different LPPMs.
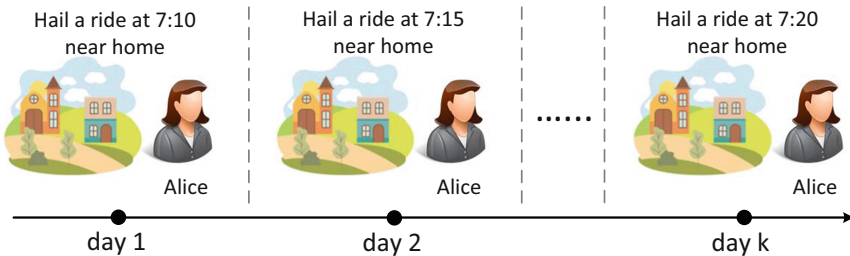
Fig. 1. A rider Alice frequently hails a ride near her home when using a RHS.

Existing LPPMs are mainly Differential Privacy (DP)-based approach [5, 9, 27, 30] and cryptography-based approach [4, 7, 18]. The first one samples a random noise from a distribution (e.g., Laplace) and adds it to the location. This approach can be proven to be differentially private. The second one utilizes homomorphic encryption [7], private set intersection [4], and secure searchable encryption [18] to process locations, such that adversaries only have a negligible

advantage on differentiating locations. Unfortunately, they cannot be applied to protect the true locations of riders in RHS. First, the DP-based schemes may output an odd location for the user to reach and they did not capture the mobility similarity between different users. Second, the cryptography schemes enforce too many computational burdens on users or server. Third, they did not consider the *semantic meanings* of locations for RHSs. It is shown that revealing information about the semantic type of locations can reduce geographical location privacy by 50% [6].

1.1 Motivations

Our motivation arises from achieving *location indistinguishability* and *semantic indistinguishability* of pick-up locations while defending the location inference attacks and the membership inclusion attack. Location indistinguishability refers to the new objective that a user's true location is indistinguishable from a group of nearby riders who share a similar mobility pattern. Semantic indistinguishability indicates that submitted locations from the same rider do not leak semantic information of the true location. We note that the semantic indistinguishability where is different from the one in modern cryptography [16] that applies to encryption domain.

1.2 Technical Challenges and Proposed Solution

To achieve our goals, we have two technical challenges to solve. **Challenge 1.** We should consider the mobility pattern of users who hail rides in the same area and find the ones who share the same mobility pattern with the target rider. How to calculate the mobility similarity of users is the first challenge. **Challenge 2.** We should define a specific region covering the riders above make the recommended location semantically different from the true location. How to calculate semantic distance between two locations is the second challenge. In summary, we need to make sure that a recommended location is indeed location indistinguishable as well as semantically dissimilar.

Intuitively, we recommend a set of pick-up locations in the following steps. First, we model the rider mobility (which reflects how a user moves in a city) as a time-dependent first-order Markov chain [26,28] on a set of locations. This is because users have a certain pattern of moving and it correlates with time. A rider's mobility profile is a transition probability matrix of the Markov chain related to the rider's mobility and visiting probability distribution over locations [8]. Second, we calculate the geographic similarity between riders by using the Mallows distance [17] and classify them into different geographic groups. We assume that there are at least k riders in each ride-hailing area, which constitutes k-anonymity (an adversary cannot distinguish a target user from other $k - 1$ users), but with stronger protection for absorbing their similar mobility pattern. We calculate the overlapping area of each alternative rider and the current rider. The resulted set of areas is prepared for finding semantically dissimilar locations later. Third, we extract location semantics through

user generated contents (UGCs) from social networks and obtain the semantic representations of locations. The UGCs include business time, rating, and type. All the contents are collected from Gaode Map [1] and Google Maps [3] are preprocessed. We use cosine similarity [14,29] to compute individual semantic similarities between locations from heterogeneous cues and fusion them in a unified hypergraph framework [15] to compute the semantic similarities between locations. Finally, we output a set of recommended pick-up locations. Riders can choose one location from the set to request their ride. To evaluate the performance of the proposed scheme, we build our mobility model upon a real-world dataset GeoLife [2,11], and leverage the walking distance function and waiting time to show its utility.

1.3 Paper Organization

The remaining of this paper proceeds as below. We review some related work in Sect. 2. We elaborate on the system model, threat model, and design objectives in Sect. 3. We present the proposed scheme in Sect. 4. We formally analyze the privacy of the scheme in Sect. 5. In Sect. 6, we implement the system and analyze its performance. Lastly, we provide some discussions in Sect. 7 and conclude this paper in Sect. 8.

2   Related Work

2.1   General LPPMs

Shokri et al. [27] provided a formal framework for the analysis of LPPMs. Specifically, they provide a generic model to formalize inference attacks on location-information and evaluate the performance of such attacks. Next, they design and justify the metric to quantify location privacy. A location-privacy meter is proposed to evaluate the effectiveness of various LPPMs. They also show the inappropriateness of entropy and k-anonymity. Andrés et al. [5] proposed a formal definition of location privacy geo-indistinguishability, to protect users' locations, while enabling approximate information to be collected for obtaining location-based services. Such a definition formalizes the concept of preserving users' locations within a radius R with a privacy level. Its core idea is that, for any $R > 0$, the user has ϵR privacy within R, i.e., the privacy level is proportional to R. Cao et al. [9] extend differential privacy to ϵ-spatiotemporal event privacy by formally defining spatiotemporal event as Boolean expressions between location and time predicates. They design a framework to transform an existing LPPM into one preserving spatiotemporal event privacy against adversaries with any prior knowledge.

2.2   LPPM for Meeting Location Determination

There is some related work on protecting locations when a meeting location is to be determined. Bilogrevic et al. [7] formulate the Fair Rendez-Vous Point (FRVP) problem for a group of users as an optimization problem and propose two algo-

rithms based on homomorphic cryptosystems for solving the FRVP problem in a privacy-preserving way. Each user provides only a single location preference to a server. However, this approach brings extra communication costs to users and computational costs to the server. Aïvodji et al. [4] utilized privacy-enhancing technologies and multimodal shortest path algorithms to compute meeting points for both drivers and riders in ride-sharing services. Rider and drivers identify potential locations locally and collaboratively compute common pick-up locations via a private information retrieval method. However, it requires too much computation and communication burden onto users. Zhang et al. [30] designed a location privacy protection scheme ShiftRoute for navigation services. It enables users to query a route without disclosing any meaningful location information. Its main idea is to selectively shift the start point/endpoint to the ones close-by and guarantee that the semantic meanings of the two points change much but preserve service usability. However, ShiftRoute only defines a simple semantic distance function with two outputs 0 and 1, which are far from enough.

3   Problem Formulation

3.1   System Model

Different from a typical system model of RHS, which consists of rider, driver, and RHSP, our system model mainly focuses on the rider. We aim to recommend pick-up locations locally on the rider side, and we do not rely on another party to compute the locations. The rider is a user requesting a ride on the roadside by sending a ride request via a smartphone application to the RHSP. The original ride request includes a true (current) location and a destination. The true location is assumed to be a frequently visited location of the rider, e.g., home and work. Each rider has an acceptable walking distance $wDis$ and an acceptable waiting time wt of location recommendation. Here, the $wDis()$ is computed by invoking a walking distance computing function from Gaode. After the rider inputs a true location tl and a destination de, the application will automatically calculate a set of recommended locations for the rider to choose from. The key notations are listed in Table 1.

3.2   Threat Model

The privacy threat is raised from the honest-but-curious RHSP and passive adversaries observing from outside of the system. It is against the users whose trajectories are sampled in our algorithm. In this case, the adversary knows that all submitted locations are generated. His attack agenda is to extract location or semantic information about the true locations of users.

3.3   Design Objectives

We have three design objectives for recommending a pick-up location: location indistinguishability and semantic indistinguishability, while not sacrificing utility.

Table 1. Key notations

Notations	Meaning
tl, de	True location, destination
rl	Recommended location
$\mathcal{ML}$	Set of marked locations on the map
$\mathcal{RL}$	Set of recommended locations
K	Number of locations in $\mathcal{RL}$
$\mathcal{C}$	Textual description repository
$wDis$	Walking distance between two locations
wt	Waiting time of location recommendation
Sim_g	Geographic similarity between two riders
Sim_s	Semantic similarity between two locations
α	Required minimum of geographic similarity
β	Required maximum of semantic similarity

– **Location indistinguishability.** We need to guarantee location indistinguishability between (1) the current rider and his/her nearby $k-1$ riders and (2) the recommended locations and their true location underneath. We use a function $\mathsf{Sim}_g(l_i, l_j)$ to measure the geographic similarity between two locations l_i, l_j. We require that the geographic distance between the recommended location rl and the true location tl is bigger than α, i.e., $\mathsf{Dis}(rl, tl) > \alpha$. We give a formal definition of location indistinguishability as follows.

Definition 1 (Location indistinguishability). *Given k riders with a similar mobility pattern, an adversary $\mathcal{A}$ cannot distinguish 1) a rider r_i from the other $k-1$ riders, and 2) a recommended location l from the true location tl, i.e.,*

$$|\Pr[\mathcal{A}(l_{r_i}) = r_i] - \Pr[\mathcal{A}(l_{r_j}) = r_i]| \leq \mathrm{negl}(k), j \in [1, i-1] \wedge [i, k],$$
$$|\Pr[\mathcal{A}(l) = tl] - \Pr[\mathcal{A}(tl) = tl]| \leq \mathrm{negl}(k).$$

– **Semantic indistinguishability.** Besides the location indistinguishability, we have to consider the semantic meaning of pick-up locations such that the semantic of true location cannot be acquired from one of recommended locations. We define a function $\mathsf{Sim}_s(l_i, l_j)$ to measure the semantic similarity between two locations l_i, l_j. We require that the semantic similarity between the recommended location rl and the true location tl is small than β, i.e., $\mathsf{Dis}(rl, tl) < \beta$. We give a formal definition of semantic indistinguishability as follows.

Definition 2 (Semantic indistinguishability). *Given a recommended location l, its true location tl, and a semantic function $Sim()$, an adversary $\mathcal{A}$ cannot distinguish l from tl, i.e.,*

$$|\Pr[\mathcal{A}(l, Sim(l)) = tl] - \Pr[\mathcal{A}(tl, Sim(tl)) = tl]| \leq \mathrm{negl}(k).$$

– **Utility.** Even though we aim to protect the location privacy of riders, we cannot ignore utility. We do not want riders to walk too far away from their true locations, which is the walking distance $wDis(rl, tl)$ between the recommended location rl and the true location tl. In addition, to guarantee user experience, we also have to control the local computational costs so that the recommendation process does not incur too much waiting time wt for riders. Given that different users may have different requirements on utility, $wDis$ and wt can vary according to their own choices.

4   Proposed Scheme

In this section, we first give an overview of our proposed scheme and then present the detailed steps.

4.1   Overview

We now provide an overview of our scheme in Fig. 2. In step 1, we model the rider mobility as a time-dependent first-order Markov chain [26,28] on a set of locations. In step 2, we obtain the mobility models of all riders. In step 3, we compute the mobility similarity between riders and build a location similarity graph of riders. In step 4, we compute the location distance between riders and choose $k-1$ riders. Till here, we have acquired the geographic similarity between riders. In step 5, we comprehend the location semantics from heterogeneous UGCs. In step 6, we compute the semantic similarity between marked locations on the map and build a similarity graph of locations. In step 7, we compute the semantic distance between locations and choose from dissimilar locations. Finally, we recommend a set of locations to the rider. We provide the details of our recommendation scheme in Algorithm 1.
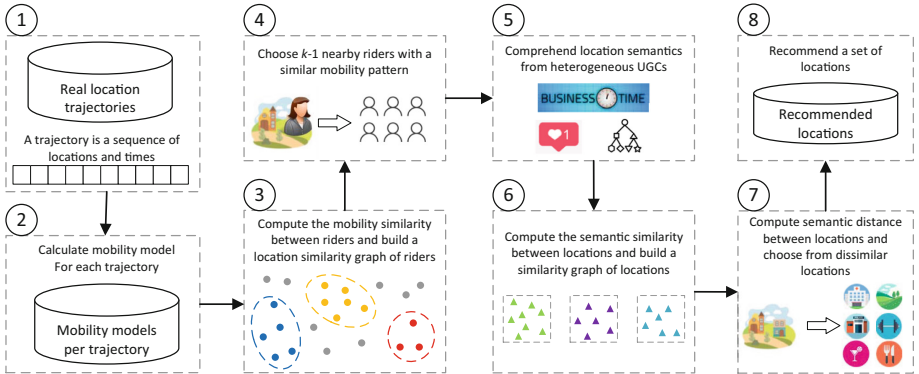

Fig. 2. Overview of recommendation algorithm.

Algorithm 1: Recommendation Algorithm

Input: r_i, tl, de
Output: $\mathcal{RL}$
//Read Riders' Trajectories
Store a rider's trajectory as a sequence of locations and times;
//Calculating Mobility Model
for $(j = 1; j \leq n; j++)$ **do**
$\quad$ Model the rider mobility as a time-dependent first-order Markov chain;
$\quad$ Compute the r_i's mobility profile $\langle p(r_i), g(r_i) \rangle$;
//Computing Mobility Similarity
for $(j = 1; j \leq n \wedge j \neq i; j++)$ **do**
$\quad$ Compute $\mathbb{E}[M_d(p_{l,t}^{l',t'}(r_i), p_{l,t}^{l'',t'}(r_j))]$;
$\quad$ Compute $\mathsf{Sim}_g(r_i, r_j) = 1 - \frac{\mathbb{E}[M_d(p_{l,t}^{l',t'}(r_i), p_{l,t}^{l'',t'}(r_j))]}{con}$;
//Rider Section
Select $k-1$ riders with the smallest Sim_g with r_i;
//Comprehending Location Semantics
Form a minimum circle $\mathcal{C}$ covering the k users;
Collect a location set $\mathcal{ML}$ of all the marked locations in $\mathcal{C}$;
for $(i = 1; i \leq |\mathcal{ML}|; i++)$ **do**
$\quad$ Compute three semantic vectors for l_i from $\mathcal{B}, \mathcal{R}, \mathcal{T}$;
//Computing Semantic Similarity
for $(i = 1, j = 1; i, j \leq |\mathcal{ML}|, i \neq j; i++, j++, \mathcal{D} = \mathcal{B}, \mathcal{R}, \mathcal{T})$ **do**
$\quad$ Compute $\mathsf{Sim}_s^{\mathcal{D}}(l_i, l_j) = \frac{\mathbf{v}_{l_i}^{\mathcal{D}} \cdot \mathbf{v}_{l_i}^{\mathcal{D}}}{||\mathbf{v}_{l_i}^{\mathcal{D}}|| ||\mathbf{v}_{l_j}^{\mathcal{D}}||}$;
//Computing Semantic Distance
Construct a hypergraph to compute $\mathsf{Sim}_s(l_i, l_j) = f[j]$;
Form K groups based on their semantic distances;
//Recommending Locations
Randomly choose K locations from K groups;
Insert the chosen locations to $\mathcal{RL}$;
return $\mathcal{RL}$;

4.2   Modeling Rider Mobility

We model the rider mobility as a time-dependent first-order Markov chain on a set of locations. A rider's mobility profile $\langle p(r), g(r) \rangle$ is a transition probability matrix of the Markov chain related to the rider's mobility and visiting probability distribution over locations. Specifically, $p_{l,t}^{l',t'}(r)$ of $p(r)$ is the probability that rider r will move to location l' in the next time instant t' when r is now at l. $g_{l,t}(r)$ is the probability that r is in l in time period t.

4.3   Calculating Mobility Similarity

Assume that now we are to recommend a set of pick-up locations for a current rider r at a true location tl. We compute the mobility similarity as follows.

– The geographic similarity captures the correlation between trajectories that are generated by two rider's mobility profiles. It indicates whether two riders visit similar locations over time with similar probabilities and if they move between those locations also with similar probabilities [8]. We compute the geographic similarity of two riders based on Mallows distance. The dissimilarity of two mobility profiles $\langle p(r), g(r) \rangle$ and $\langle p(s), g(s) \rangle$ is defined as the Mallows distance of the next random locations l' and l'':

$$\mathbb{E}[M_d(p_{l,t}^{l',t'}(r), p_{l,t}^{l'',t'}(s))], \tag{1}$$

where d is an arbitrary distance function and the expectation is calculated over random variable l and time periods l and l'.

– The geographic similarity between two mobility patterns of r and s is defined as:

$$\mathsf{Sim}_g(r,s) = 1 - \frac{\mathbb{E}[M_d(p_{l,t}^{l',t'}(r), p_{l,t}^{l'',t'}(s))]}{con}, \tag{2}$$

where con is a constant ensuring that the Sim_g stays between 0 and 1.

Next, we compute the geographic similarity between r and other riders, and select the nearby $k-1$ riders with a similar mobility pattern to the current rider as depicted in Fig. 3. As new riders and their trajectories join, we will update the circle and renew the location pool.
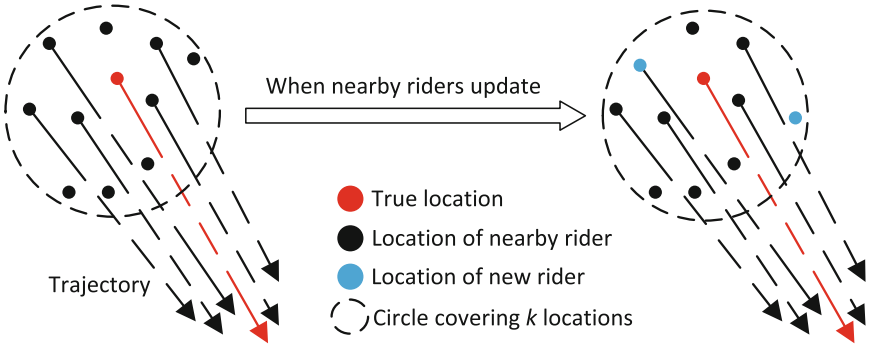
Fig. 3. Circling k riders with similar mobility pattern and updating the circle.

4.4   Comprehending Location Semantics

Given the other $k-1$ riders, we collect the $k-1$ locations from the $k-1$ users that are near tl and form a minimum circle that is covering all the k users. Next, we collect a location set of all the marked locations $\mathcal{ML}$ in the circle on the map, e.g., supermarket, shoe store, barber shop, and restaurant. Since the heterogeneous UGCs capture semantics of a location, we comprehend them in $\mathcal{ML}$ from four aspects: business time, rating, and type.

– Business time is shown for an open venue, e.g., "08:00-16:00" for a Tim Hortons coffee shop. Different business time indicates the type of a location to some extent. We quantize the business time into 48 time zones a day, i.e., half an hour for each time zone. For each location l_i, the business time is separated and put into the corresponding bin. The business time distribution vector of a location venue is the histogram of business time associated to the location

$$\mathbf{v}_{l_i}^{\mathcal{B}} = [f_{l_i}^{\mathcal{B}}(1), f_{l_i}^{\mathcal{B}}(2), ..., f_{l_i}^{\mathcal{B}}(48)], \tag{3}$$

where $f_{l_i}^{\mathcal{B}}(j)$ denotes the openness of the location at time j, i.e., 1 means open and 0 otherwise. However, we observe that not all locations are open to public, e.g., a commercial building/university that requires an entrance guard card/student ID card to enter. For these locations, we manually mark their business time distribution vector as $[0, 0, ..., 0]$.

– Rating r_{l_i} reveals customers' opinions toward a location. They are usually adopting the five-star rating mechanism on many social networks. We extract the ratings and generate a rating vector

$$\mathbf{v}_{l_i}^{\mathcal{R}} = [f_{l_i}^{\mathcal{R}}(1), f_{l_i}^{\mathcal{R}}(2), ..., f_{l_i}^{\mathcal{R}}(5)], \tag{4}$$

where $f_{l_i}^{\mathcal{R}}(5 - j) = 1$ for $0 \le j < 5$ if $j < r_{l_i}$ and $f_{l_i}^{\mathcal{R}}(5 - j) = 0$ otherwise. For example, $\mathbf{v}_{l_i}^{\mathcal{R}} = [0, 1, 1, 1, 1]$ when $r_{l_i} = 4$.

– Type shows the classification of a location. We use the POI classification method from Gaode Maps [1] and extract the ratings and generate a type vector

$$\mathbf{v}_{l_i}^{\mathcal{T}} = [f_{l_i}^{\mathcal{T}}(1), f_{l_i}^{\mathcal{T}}(2), ..., f_{l_i}^{\mathcal{T}}(23)], \tag{5}$$

where $f_{l_i}^{\mathcal{T}}(j)$ denotes whether the location belongs to type j, i.e., 1 means yes and 0 otherwise.

4.5   Calculating Semantic Similarity

We compute the semantic similarity between locations as follows.

– We use the cosine similarity metric to compute the similarity of two locations at each individual dimension

$$\mathsf{Sim}_s^D(l_i, l_j) = \frac{\mathbf{v}_{l_i}^{\mathcal{D}} \cdot \mathbf{v}_{l_i}^{\mathcal{D}}}{||\mathbf{v}_{l_i}^{\mathcal{D}}||||\mathbf{v}_{l_j}^{\mathcal{D}}||}, \tag{6}$$

where $\mathcal{D} = \mathcal{B}, \mathcal{R}, \mathcal{T}$. The semantic distance between two locations is denoted as $\mathsf{Dis}_s(l_i, l_j) = 1 - \mathsf{Sim}_s^D(l_i, l_j)$. $\overline{\mathsf{Dis}^{\mathcal{D}}}$ denotes the mean value of elements in the Dth distance matrix.

– Based on the hypergraph framework, we take each location venue as a centroid and collect the k-nearest neighbors. The hyperedge weight is calculated as

$$we(e) = \sum_{v_i, v_j \in e} \mathbf{A}_{ij}, \tag{7}$$

where $\mathbf{A}_{ij}$ is the affinity between vertex v_i and v_j. $\mathbf{A}_{ij} = \exp\left(-\sum_{D=1}^{3} \frac{\mathsf{Dis}_{ij}^D}{3\overline{\mathsf{Dis}^D}}\right)$.

– We construct the hypergraph for location semantics and compute the hypergraph Laplacian, then we select a query location l_i and use a query vector $y \in R^{|\mathcal{ML}|}$, only the entry corresponding to the query location is set to be 1 and all others are set to 0. After solving the linear system $(\mu I + \Delta)f = \mu y$ where Δ is the hypergraph Laplacian matrix, we obtain the ranking scores $f \in R^{|\mathcal{ML}|}$ [15].

– We perform an normalization method on ranking scores f. Finally, the similarity between the query location l_i and an other location l_j is

$$\mathsf{Sim}_s(l_i, l_j) = f[j]. \tag{8}$$

– Finally, we build a semantic similarity graph of all the marked locations and tl by classifying the locations into K groups according to their semantic similarity.

We note that we need to filter some noises when handling business time and ratings from UGCs. Some locations do not have the information of business time or rating. For these locations, we manually mark their business time as the one of the locations in their same classification that have the most probable business time. We also rate these locations as three stars as default.

4.6 Recommending a Set of Pick-Up Locations

After obtaining the semantic similarity graph, we randomly choose K locations from K different groups that have a low semantic similarity to tl and form a set of recommended locations $\mathcal{RL}$. Specifically, we set the business time of tl as "19:00-08:00" [12] and rating as the sequence of the unit price range in the city, which lays a foundation for semantic distance calculation. For example, we divide the unit prices of residence communities in Beijing into five ranges $[0, 50000]$, $[50001, 100000]$, $[150001, 200000]$, $[250001, 300000]$, and $[300001, \infty]$. If the unit price of a residence community is 120000 yuan, then it belongs to the second unit price range and its rating is $[0, 0, 0, 1, 1]$, which is similar to the rating mechanism. To satisfy the real demands of riders, we provide two metrics for them, i.e., walking distance between two locations $wDis$ and waiting time wt. We consider these two metrics during the selection of $k-1$ riders as well as K marked locations.

5 Privacy and Security Analysis

5.1 Location Indistinguishability

We model the mobility patterns for riders and compute the geographic similarity between $k-1$ nearby riders with the current rider r. Next, we classify the nearest $k-1$ riders with r. A minimum circle covering the k riders is calculated as the potential area within which we recommend a pick-up location. By doing so, an adversary cannot differentiate r from other $k-1$ riders within this circle since the

users of these true locations inside the circle all share a similar mobility pattern. Thus location indistinguishability between the current rider and his/her nearby $k - 1$ riders is achieved, i.e., $|\Pr[\mathcal{A}(l_{r_i}) = r_i] - \Pr[\mathcal{A}(l_{r_j}) = r_i]| \leq \text{negl}(k), j \in [1, i-1] \wedge [i, k]$.

Instead of generating noise and adding it to the true location, we only recommend a marked locations on the map excluding the true location. Meanwhile, we update the circle with the trajectories such that new marked locations will be added to the location pool. More importantly, we do not leak any useful information of the true location by releasing the recommended location. In this way, an adversary cannot infer the true location from the recommended locations, thus achieving location indistinguishability between the recommended locations and its true location underneath, i.e., $|\Pr[\mathcal{A}(l) = tl] - \Pr[\mathcal{A}(tl) = tl]| \leq \text{negl}(k)$.

5.2   Semantic Indistinguishability

Based on the circle we have calculated from users' mobility patterns, we comprehend the semantics of all the marked location on the map and classify them according to their semantic similarities. Locations with similar semantics are grouped and separated from the ones with dissimilar semantic meanings. Next, we only choose K locations from K different semantic groups. The semantic similarity between this group and the one to which the true location belongs to is less than a threshold β. Therefore, an adversary cannot acquire the semantic of the true location from the one observed from the recommended locations which have distant semantic meanings, achieving semantic indistinguishability, i.e., $|\Pr[\mathcal{A}(l, Sim(l)) = tl] - \Pr[\mathcal{A}(tl, Sim(tl)) = tl]| \leq \text{negl}(k)$.

## 6   Performance Evaluation

6.1   Experimental Settings

We implement the recommendation algorithm on a desktop with AMD Ryzen5 3600 CPU, 16 GB memory, and Windows 10 professional operating system. The experimental parameters and their values are listed in Table 2.

Table 2. Experimental parameters

Parameters	Value
k	12
K	5
$wDis$ (meters)	[10, 500]
wt (millisecond)	(0, 100]
α	0.7
β	0.3

6.2   Dataset

The dataset is GeoLife 1.3 [2] collected in a Microsoft project which consists of
18, 760 trajectories with 24, 876, 978 locations and 50, 186 h. The mean number
of points of each trajectory is 1,332, and the mean duration of the trajectories
is 7.26 min. As it did not explicitly mark the home or work for users, we only
choose a set of trajectories that resemble those covering home and work.
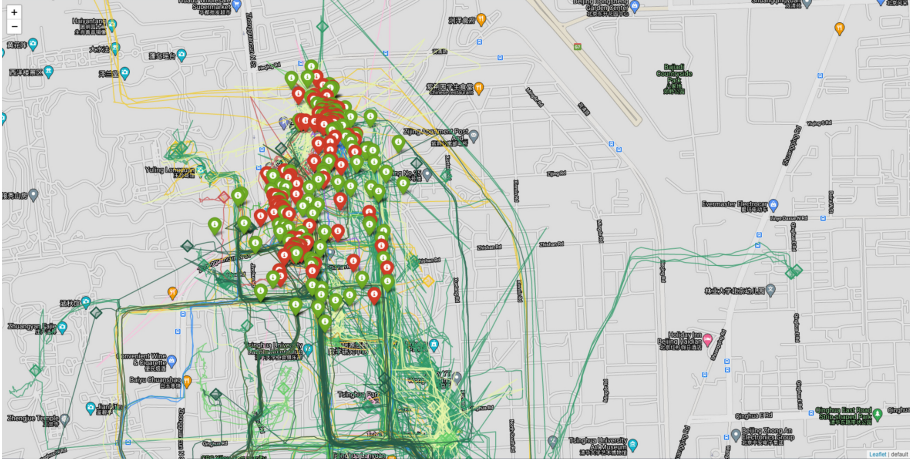

Fig. 4. Marking green start points, red endpoints, and trajectories for users. (Color
figure online)

6.3   Computational Costs

Now we analyze the computational costs of processing trajectory dataset, finding
$k-1$ riders with similar mobility pattern, and finding K locations with dissimilar
semantic at the rider side. We first cluster the star points and endpoints of
182 users by using DBSCAN [13] to observe potentially similar trajectories,
as shown in Fig. 4 and Fig. 5. It takes us 19.6 min to finish the clustering and
obtain 581 trajectory groups. Since we need to define "same location", we cluster
adjacent locations within each trajectory group. For example, we choose the #1
group with 12 users, which takes 360 ms to model their mobility patterns, i.e.,
computing their transition probability matrix.

Next, we compute all the 11 geographic similarities for all the 12 users. As
shown in Fig. 6, the time of comparing with 11 users is approximately 7 s, i.e.,
it costs less than 1 s to compute one geographic similarity for one pair of users.
Afterward, we compute a minimum circle with a radius of 500 m covering the
obtained k locations and process the semantics of all the marked locations in
the circle.
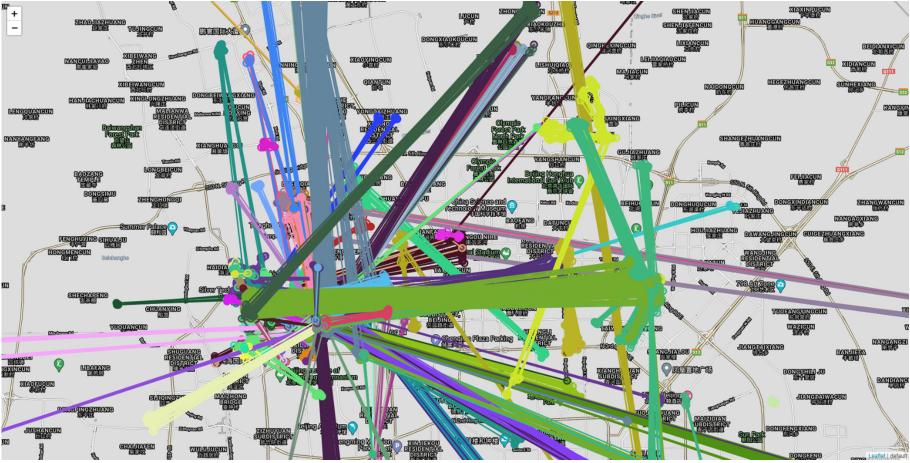
Fig. 5. Clustering start points and endpoints for users.

We find 83 marked locations within this circle. In the beginning, we need to compute the semantic similarity between the home and the remaining 82 locations. We show the time costs of 10 different home locations in Fig. 7, where the time cost is less than 0.03 ms. Automatically, the 82 locations are classified into 10 groups according to their similarity with the one of hl of a target rider. Afterwards, we can select the top K locations from K similarity groups with the smallest semantic similarity. It is to be noted that modeling the mobility patterns, computing geographic similarity, drawing the circle, and classifying locations, could be preprocessed locally for each rider, thus saving the time.
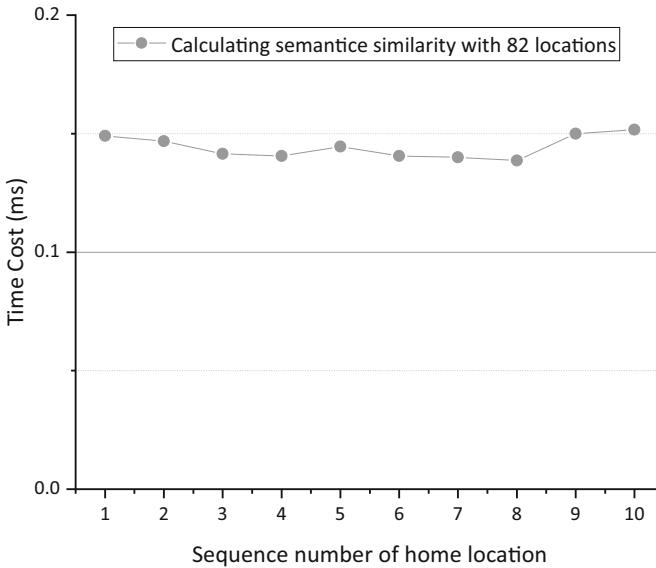
6.4   Utility

To analyze the utility, we consider two metrics for riders, namely walking distance $wDis$ and waiting time wt. We first compute the walking distance from tl to other 82 locations by querying the cloud server in the normal way. Each query takes an average of 103 ms which is considered as pre-processing time. We set the K and $wDis$ as variables and see how much time the rider has to wait for in average. If a recommended pick-up location does not coincide with the $wDis$, we select the next pick-up locations with less semantic dissimilarity. From Fig. 8, when K is fixed, the waiting time increases with $wDis$ because we will have more optional locations and it takes more time to compute the semantic similarity. When $wDis$ is fixed, the walking time also increases with K for processing more locations. The reason behind the existence of several odd points is that the corresponding two variables require more search in the location pool. Finally, the recommendation time of selecting 10 pick-up locations within 500 m of walking distance is less than 0.12 ms.
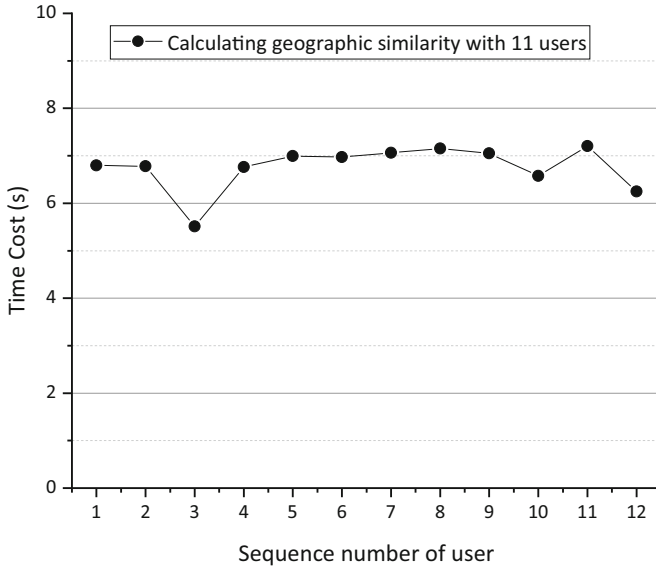
Fig. 6. Time cost in computing geographic similarity.

Fig. 7. Time cost in computing semantic similarity.

6.5   Android Implementation

We also implement our recommendation scheme on an Android smartphone. We perform preprocessing on the rider end, including modeling the mobility pat-
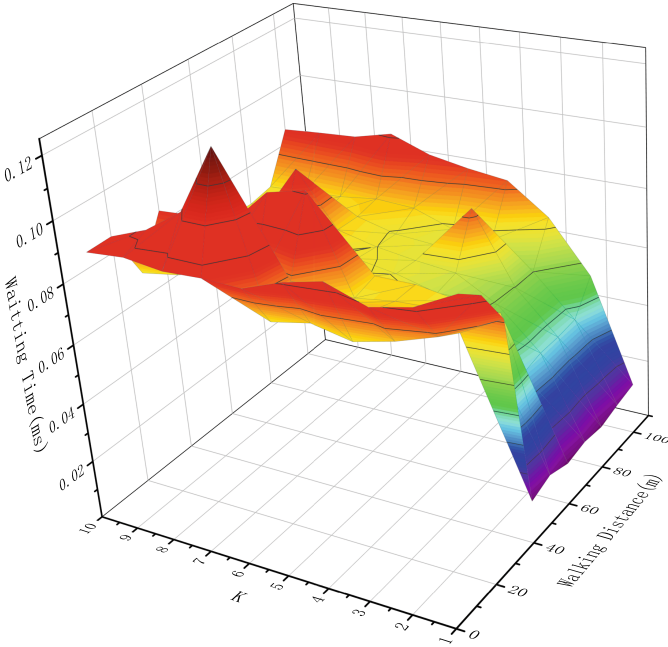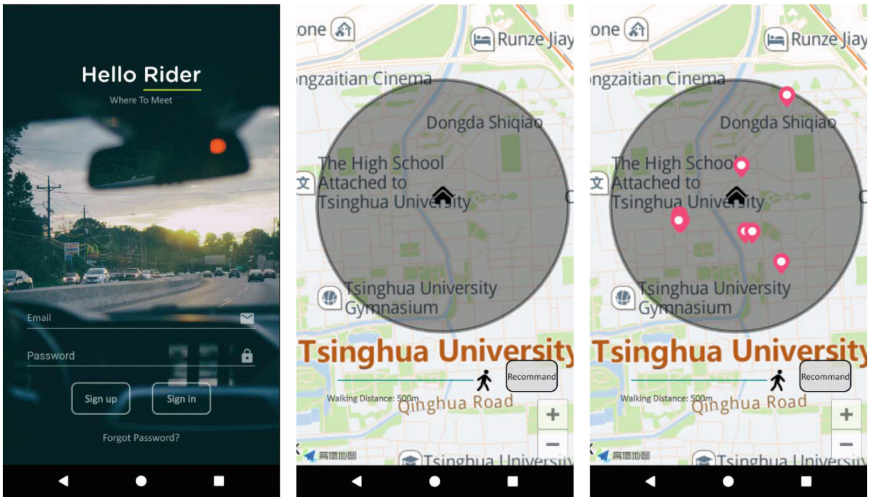
Fig. 8. Utility.

(a) Login

(b) Input pick-up location and walking distance

(c) Recommend a set of pick-up locations

Fig. 9. Android application

terns, computing geographic similarity, drawing the circle, classifying locations, and computing semantic similarity. As shown in Fig. 9, after logging in the ride-hailing app, the rider selects the home location and walking distance, and presses the recommendation button to obtain a set of pick-up locations.

7   Discussions

7.1 k-Anonymity

We assume that there are at least k riders in the ride-hailing area of the target rider, which constitutes k-anonymity. Meanwhile, we provide stronger protection for absorbing their similar mobility pattern. If this assumption does not hold, say there are not enough riders in the area, we can leverage the residence communities nearby as a backup approach.

7.2   Protection of Destination

Although the location recommendation scheme in this work is mainly designed for the pick-up location, it is also applicable to the protection of destinations. This is because the process of the two types of locations are the same since they have nearby riders and residence communities. All the considerations for pick-up locations are applicable to the destinations.

8   Conclusions and Future Work

In this work, we consider both mobility patterns and location semantics in choosing a pick-up location for ride-hailing services. A recommendation scheme with location indistinguishability and semantic indistinguishability is proposed. We model riders' mobility patterns as a time-dependent first-order Markov chain and compute the geographic similarity between riders by using the Mallows distance. We further comprehend the semantics of locations based on user generated contents from social networks and compute the semantic similarity between locations by using cosine similarity and a unified hypergraph. The experimental results over real-world dataset and an Android smartphone show that it only costs less than 0.12 ms to recommend 10 pick-up locations within 500 m of walking distance.

The future work is aimed at 1) further exploiting the theoretic aspects of the proposed scheme, and 2) experimenting on a large scale dataset to evaluate the efficacy and efficiency of the algorithm.

References

1. Gaode Map. https://lbs.amap.com. Accessed 15 Apr 2021
2. GeoLife GPS Trajectories. https://www.microsoft.com/en-us/download/details.aspx?id=52367. Accessed 15 Apr 2021
3. Google Maps. https://developers.google.com/maps. Accessed 15 Apr 2021
4. Aïvodji, U.M., Gambs, S., Huguet, M.J., Killijian, M.O.: Meeting points in ridesharing: a privacy-preserving approach. Transp. Res. Part C **72**, 239–253 (2016)
5. Andrés, M.E., Bordenabe, N.E., Chatzikokolakis, K., Palamidessi, C.: Geo-indistinguishability: differential privacy for location-based systems. In: Proceedings of 20th ACM Conference on Computer and Communications Security (CCS), Germany, pp. 901–914, November 2013
6. Ağır, B., Huguenin, K., Hengartner, U., Hubaux, J.P.: On the privacy implications of location semantics. In: Proceedings of 16th Privacy Enhancing Technologies (PETS), pp. 165–183, October 2016
7. Bilogrevic, I., Jadliwala, M., Joneja, V., Kalkan, K., Hubaux, J.P., Aad, I.: Privacy-preserving optimal meeting location determination on mobile devices. IEEE Trans. Inf. Forensics Secur. (TIFS) **9**(7), 1141–1156 (2014)
8. Bindschaedler, V., Shorki, R.: Synthesizing plausible privacy-preserving location traces. In: Proceedings of 37th IEEE Symposium on Security and Privacy (S&P), pp. 546–563, May 2016
9. Cao, Y., Xiao, Y., Xiong, L., Bai, L., Yoshikawa, M.: Protecting spatiotemporal event privacy in continuous location-based services. IEEE Trans. Knowl. Data Eng. (TKDE) **99**, 1–13 (2019)
10. Chen, Y., Li, M., Zheng, S., Hu, D., Lai, C., Conti, M.: One-time, oblivious, and unlinkable query processing over encrypted data on cloud. In: Proceedings of 22nd International Conference on Information and Communications Security (ICICS), Copenhagen, Denmark, pp. 350–365, August 2020
11. Chen, Z., Shen, H.T., Zhou, X., Zheng, Y., Xie, X.: Searching trajectories by locations: an efficiency study. In: Proceedings of 29th ACM SIGMOD International Conference on Management of Data (SIGMOD), Indiana, USA, pp. 255–266, June 2010
12. Drakonakis, K., Ilia, P., Ioannidis, S., Polakis, J.: Please forget where i was last summer: the privacy risks of public location (meta)data. In: Proceedings of 26th Annual Network and Distributed System Security Symposium (NDSS), USA, February 2019
13. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD), Portland, USA, pp. 226–231, August 1996
14. Huang, A.: Similarity measures for text document clustering. In: Proceedings of Sixth New Zealand Computer Science Research Student Conference (NZCSRSC), Christchurch, New Zealand, pp. 49–56 (2008)
15. Huang, Y.: Hypergraph based visual categorization and segmentation. Ph.D. thesis, Rutgers Univ., New Brunswick, USA (2010)
16. Katz, J., Lindell, Y.: Introduction to Modern Cryptography, 2nd edn. Chapman and Hall/CRC (2014)
17. Levina, E., Bickel, P.: The earth mover's distance is the mallows distance: some insights from statistics. In: Proceedings of 8th IEEE International Conference on Computer Vision (ICCV), Vancouver, Canada, pp. 251–256 (2001)

18. Li, M., Chen, Y., Zheng, S., Hu, D., Lal, C., Conti, M.: Privacy-preserving navigation supporting similar queries in vehicular networks. IEEE Trans. Dependable Secure Comput. (TDSC) **99**, 1–16 (2020). https://doi.org/10.1109/TDSC.2020.3017534

19. Li, M., Gao, J., Chen, Y., Zhao, J., Alazab, M.: Privacy-preserving ride-hailing with verifiable order-linking in vehicular networks. In: Proceedings of 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Guangzhou, China, pp. 599–606, December 2020

20. Li, M., Zhu, L., Lin, X.: CoRide: a privacy-preserving collaborative-ride hailing service using blockchain-assisted vehicular fog computing. In: Proceedings of ACM 15th EAI International Conference on Security and Privacy in Communication Networks (SecureComm), Orlando, USA, pp. 408–422, October 2019

21. Li, M., Zhu, L., Lin, X.: Privacy-preserving traffic monitoring with false report filtering via fog-assisted vehicular crowdsensing. IEEE Trans. Serv. Comput. (TSC) **99**, 1–11 (2019). https://doi.org/10.1109/TSC.2019.2903060

22. Li, M., Zhu, L., Zhang, Z., Xu, R.: Differentially private publication scheme for trajectory data. In: Proceedings of 1st IEEE International Conference on Data Science in Cyberspace (DSC), Changsha, China, pp. 596–601, June 2016

23. Li, M., Zhu, L., Zhang, Z., Xu, R.: Achieving differential privacy of trajectory data publishing in participatory sensing. Inf. Sci. **400–401**, 1–13 (2017). https://doi.org/10.1016/j.ins.2017.03.015

24. Mazareanu, E.: Monthly number of ubeŕs active users worldwide from 2017 to 2020, by quarter (in millions) (2020). https://www.statista.com/statistics/833743/us-users-ride-sharing-services. Accessed 15 Apr 2021

25. Pham, A., Dacosta, I., Endignoux, G., Troncoso-Pastoriza, J., Huguenin, K., Hubaux, J.P.: ORide: a privacy-preserving yet accountable ride-hailing service. In: Vancouver, C. (ed.) Proceedings of 26th USENIX Security Symposium (USENIX Security), pp. 1235–1252 (2017)

26. Sahina, A.D., Sen, Z.: First-order Markov chain approach to wind speed modelling. J. Wind Eng. Ind. Aerodyn. **89**, 263–269 (2001)

27. Shokri, R., Theodorakopoulos, G., Boudec, J.Y.L., Hubaux, J.P.: Quantifying location privacy. In: Proceedings of 32th IEEE Symposium on Security and Privacy (S&P), Oakland, USA, pp. 247–262, May 2011

28. Tan, C.C., Beaulieu, N.C.: On first-order Markov modeling for the Rayleigh fading channel. IEEE Trans. Commun. **48**(12), 2032–2040 (2000)

29. Wang, X., et al.: Semantic-based location recommendation with multimodal venue semantics. IEEE Trans. Multimed. (TMM) **17**(3), 409–419 (2015)

30. Zhang, P., Hu, C., Chen, D., Li, H., Li, Q.: ShiftRoute: achieving location privacy for map services on smartphones. IEEE Trans. Veh. Technol. (TVT) **67**(5), 4527–4538 (2018)

31. Zhu, L., Li, M., Zhang, Z., Qin, Z.: ASAP: an anonymous smart-parking and payment scheme in vehicular networks. IEEE Trans. Dependable Secure Comput. (TDSC) **17**(4), 703–715 (2020). https://doi.org/10.1109/TDSC.2018.2850780