Journal of Network and Computer Applications 53 (2015) 57–73

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

A fast privacy-preserving framework for continuous location-based queries in road networks ☆

Yong Wang [a], Yun Xia [a], Jie Hou [a], Shi-meng Gao [a], Xiao Nie [a], Qi Wang [b,*]

[a] School of Computer Science and Engineering, University of Electronic Science and Technology of China, 611731 Chengdu, China
[b] National computer network emergency response technical team/coordination center of China, 100190, Beijing, China

ARTICLE INFO

Article history:
Received 4 December 2013
Received in revised form
25 August 2014
Accepted 31 January 2015
Available online 14 March 2015

Keywords:
Privacy-preservation
Location based services (LBS)
Road networks
Continuous query

ABSTRACT

The prevalence of location based services (LBS) gives rise to personal privacy concerns as users share their locations and queries to obtain desired services. For continuous queries where users report their locations periodically, attackers can infer more about users' privacy by analyzing the correlations of their snapshot samples. Traditional privacy-preserving solutions designed in Euclidean space can be hardly applied to the road network environment because of their ignorance of network topological properties. In this paper, we propose a novel continuous query privacy-preserving framework in road networks. Our framework is based on the concepts of *k*-anonymity and *l*-diversity. To achieve the quality of service, the distance limitation is taken into account. We build an *Snet* hierarchy based on the density of users, history traces, and road network topologies to accelerate the cloaking process performed at the anonymization server. Two types of cloaking algorithms, for a single user and a batch of users, are designed. The security analysis shows that our framework is robust to typical attacks. We evaluate our framework from the aspects of privacy-preserving ability, quality of service, and system performance, which indicates that our framework can provide good privacy protection while ensuring users' quality of service.

© 2015 Elsevier Ltd. All rights reserved.

1. Introduction

Pushed by the widespread use of positioning devices (e.g., GPS), location-based services (LBS) have become ubiquitous in recent years. With locations (latitudes and longitudes) obtained from these devices, LBS applications can provide users with highly personalized services, through local business searches (e.g., searching for restaurants nearest to a user), e-marketing (e.g., sending e-coupons to nearby potential customers), and social networking (e.g., a batch of friends sharing their geo-tagged photos), etc. Generally, users can send two types of queries to LBS providers: *snapshot query*, for example, "Show me the hotels within one mile", and *continuous query*, for example, "Inform me of the nearest petrol station every 5 min in the next 30 min". Virtually, a continuous query consists of several consecutive snapshots, which are processed with user's real-time locations one by one.

However, as locations are reported to a potentially untrustworthy LBS provider, attackers may track users by exploiting their exposed locations, which may lead to the concern of *location privacy*. The disclosure of a user's location may reveal sensitive information, such as health condition and religious faith. In particular, such tracking capabilities of attackers trigger crime possibilities, such as vehicle theft and kidnapping. In other aspects, a user may not want to be identified as the subscriber of a specific location-based service, especially when the service is sensitive (e.g., querying for the nearest Cancer Treatment Center), which is concerned as *query privacy*. Apparently, privacy-preserving in continuous query is more challenging than that in snapshot query since an attacker could infer a user's privacy by utilizing the spatial and temporal correlations of snapshot samples. Hence, privacy of continuous queries is what we focus on.

Plenty of privacy-preserving techniques (Samarati and Sweeney, 1998; Gruteser and Grunwald, 2003; Liu et al., 2009) designed for Euclidean space have been proposed, wherein users can move in arbitrary directions at random speed. However, a user's movement may be constrained by the underlying road network. For example, a user should move along a certain road within the maximum speed limitation. Applying these techniques directly to road networks may result in privacy leakage. As shown in Fig. 1(a), *u* is anonymized with 4 other users, denoted by red points, and his exact position is blurred into a gray region with the spatial cloaking methods (Bamba et al., 2008; Gedik and Liu, 2008; Kainis et al., 2007). With such a cloaked

☆This work was supported by the Joint Funds of the National Natural Science Foundation of China (Grant no. U1230106), and by the National Information Security 242 Project of China (Grant no. 2013A050).

* Corresponding author.
E-mail address: cla@uestc.edu.cn (Y. Wang).

http://dx.doi.org/10.1016/j.jnca.2015.01.004
1084-8045/© 2015 Elsevier Ltd. All rights reserved.

region, 5-anonymity is achieved, the attacker can only tell that u might be somewhere in the gray area. Figure 1(b) shows the same case but with the knowledge of underlying road networks. Since the gray area contains a single road segment, the attacker can infer that u must be located in the road segment and users being out of the segment will be excluded. Hence, 5-anonymity is violated with only two available users, which may enable attackers to track down u much easier. Generally, this kind of attack is impossible to be applied in practice by taking underlying road networks into account while anonymizing. Furthermore, other road network properties, such as the population density, which has a significant impact on the privacy preservation, should also be concerned.

Currently, several privacy-preserving solutions have been introduced to road networks. Unfortunately, existing approaches that apply a traditional cloaking algorithm in road networks incur a huge time cost. To avoid such huge time cost in the traditional approaches, we improve the speed of retrieving users to be cloaked together based on a hierarchy structure. Furthermore, alternatively, the cloaking performs faster to process a batch of user simultaneously instead of processing a single user at one time. We believe that this is the first work to propose a fast LBS continuous query privacy preservation framework in road networks. The query privacy of a user is preserved even if his location is leaked. The network topological properties are deliberated, so that we can effectively provide privacy preservation for users while lowering down computation overheads for both of LBS providers and the privacy-preserving system. The main idea of our solution is to abstract the underlying road network into multiple levels. The abstracted unit is denoted as an *Snet*. Correspondingly, we propose an *Snet* merging algorithm to construct the *Snet* hierarchical structure (see Section 3.3.1). Based on the *Snet* hierarchy, our framework introduces a trusted third party to cloak the query issuer with others, which satisfies his specified privacy requirements (see Section 3.2). From the view of LBS providers, they can only relate a set of users to a set of queries instead of a query to a particular user.

We present two versions of privacy-preserving algorithms, one processes each query respectively, while the other handles a batch of queries simultaneously. Our main contributions include:

- The framework can resist the attacks that break k-anonymity by considering the topological properties of road networks. To accelerate the privacy-preserving process, we abstract the road network into a hierarchy structure by considering the density of users, history traces, and the connectivity of road segments.
- The whole procedure is divided into three stages: initialization stage, execution stage, and update stage. The initialization stage builds a hierarchy structure to facilitate the cloaking process. Based on pre-computed hierarchy structure, the framework can provide more efficient privacy preservation services in the execution stage. As the underlying network may change over time, the update stage enables our framework to adapt to different road conditions and maintain long-term effectiveness.
- We propose fast cloaking algorithms based on the hierarchical structure for a single user and a batch of users. Each *Snet* is treated as a cloaking unit. When users in the sub-*Snets* cannot satisfy cloaking requirements, the cloaking process will shift to the parent *Snet*.
- Users' moving trend, velocity difference, and distance difference are taken into consideration, so as to maintain as many common users as possible to resist typical attacks. The attack resilience analysis and performance evaluation indicate that our framework can resist typical attacks and achieve good performance.

The rest of our paper is organized as follows: in Section 2, we discuss related work on privacy preservation. We present the system model in Section 3, and detailed algorithms and the framework maintenance are shown in Section 4. We analyze the security of our cloaking algorithms in Section 5. Experiments and evaluations are presented in Section 6. In Section 7, we draw some brief conclusions.

2. Related work

Section 2.1 reviews related work on privacy preservation in Euclidean space, Section 2.2 surveys the literature on privacy preservation in road networks, Section 2.3 explains the privacy preservation techniques based on multiparty computation, and Section 2.4 discusses privacy preservation against typical attacks.

2.1. Privacy preservation in Euclidean space

Previous work in Euclidean space can be classified into two categories according to the system architecture: centralized privacy-preserving architecture and distributed privacy-preserving architecture.

2.1.1. Centralized privacy-preserving architecture

In the centralized privacy preservation architecture, a trusted third party is involved to blur users' locations into spatial regions, which guarantees to satisfy the k-anonymity (Samarati and Sweeney, 1998) requirement. Based on the idea of k-anonymity, the Interval Cloak algorithm (Gruteser and Grunwald, 2003) was proposed, which recursively partitions an area into four sub-areas until users in the sub-area are less than k. The centralized architecture has been applied to continuous queries (Chow and Mokbel, 2007; Wang et al., 2012a,b; Guha et al., 2012). The *L2P2* scheme was presented by Wang et al. (2012a), which allows users to define their dynamic and diverse privacy requirements for continuous queries. Wang et al. (2012b) proposed a query linking privacy-preserving algorithm (*V-DCA*) for continuous LBS queries, which considers users' velocities and acceleration similarities to select users that can stay close in the long run.

2.1.2. Distributed privacy-preserving architecture

In the distributed architecture, users protect their privacy by working collaboratively (Domingo-Ferrer, 2006) or autonomously (Olumofin et al., 2010; Huang and Vishwanathan, 2010; Durr et al., 2011). Domingo-Ferrer (2006) proposed a collaborative algorithm, in which a user broadcasts his perturbed location to form a group with $k-1$ neighbors. Olumofin et al. (2010) combined the cloaking with Private Information Retrieval (PIR). Durr et al. (2011) proposed a position sharing scheme to hide the exact location information. For continuous queries, Pingley et al. (2011) generated dummy queries based on query contexts and motion modes. Wang et al. (2012c) designed a distributed architecture with several semi-honest anonymizing servers.

Unfortunately, these are designed for Euclidean space and cannot address the problem faced by road networks. In this respect, our proposed algorithm not only considers personalized privacy requirements and moving characteristics as in Wang et al. (2012b), but also takes the underlying road networks into account. In addition, to improve the system efficiency, our algorithm can cloak for a batch of users simultaneously.

2.2. Privacy preservation in road networks

Several privacy-preserving techniques have been proposed to protect users' privacy in road networks. Based on the type of location-based queries, these techniques can be classified into two categories: privacy preservation for snapshot location-based

queries and privacy preservation for continuous location-based queries.

2.2.1. Privacy preservation for snapshot location-based queries

PSNN and *PSRQ* techniques (Ku et al., 2007) solely rely on *Casper* (Mokbel et al., 2006), which was designed for the Euclidean space. As a result, the drawbacks of techniques for Euclidean space are inherited. Kolahdouzan and Shahabi (2004) partitioned the whole road network into small *Voronoi* regions for anonymization. In Mouratidis and Yiu (2010), the Hilbert-order was used to anonymize users with their $k-1$ neighbors. Hence, the effectiveness of the algorithm depends much on the ordering. Papadias et al. (2003) expanded the cloaked road segments until privacy requirements are achieved. To balance the processing cost and privacy preservation, Wang and Liu (2009) proposed a *X*-star based privacy-preservation framework merging neighboring queries into a newly established cloaking star (super-star). Chow et al. (2011) designed an effective shared execution paradigm. Bao et al. (2009) proposed a peer-to-peer location privacy-preserving system called *Pros*, in which a user collaborates with others to form a cloaked road segment set. However, simply applying these techniques into continuous location-based queries may suffer from the attacks correlating snapshot samples.

2.2.2. Privacy preservation for continuous location-based queries

Previous research has mainly focused on breaking the continuity of location exposure by utilizing mix-zones to change users' identification. In Freudiger et al. (2009), the mixing effectiveness of possible mix zone locations was employed to optimize the placement of mix zones. Mobimix (Palanisamy and Liu, 2011) takes multiple factors into consideration in the placement of mix zones, such as the statistical behavior of the user population. However, it pays no consideration to the network updating, which may lead to system unavailability in the long run. As the placement optimization is *NP*-hard, Liu et al. (2012) designed two heuristic algorithms to strategically select mix zone locations. In general, although mix-zones protect the privacy of continuous queries, they limit the field where users are served, which may be unacceptable for some users. Hence, our framework adopts the cloaking-based mechanism for continuous queries privacy preservation, which also considers road networks update.

2.3. PIR based privacy preservation

Methods relying on cryptographic or Private Information Retrieval (PIR) are used in location privacy preservation. Private Information Retrieval (PIR) techniques allow a user to retrieve an element of a database without the owner of that database being able to determine which element was selected (Chor et al., 1998). Generally, PIR based techniques do not require a trusted third party. Zhong et al. (2007) introduced three protocols, namely Louis, Lester and Pierre, to provide location privacy when answering *K* Nearest Neighbor (KNN) queries. Similarly, Papadopoulos et al. (2010) employed secure hardware-aided PIR to achieve strong location privacy. Ghinita et al. (2008) proposed a framework to support private location-dependent queries based on PIR techniques. Their framework does not need a trusted third party and can achieve strong privacy for snapshots of users' locations. Narayanan et al. (2011) proposed a variety of cryptographic protocols that support private proximity testing. They use "location tags" generated from the physical environment to strengthen the security of proximity testing. Li and Jung (2013) designed a suite of Privacy-preserving Location Query Protocol (PLQP) to protect users' locations privacy under the application scenario of social network services (SNS).

This category of techniques provides strong privacy protection. However, its performance, although improved by utilizing special hardwares, is still hard applicable in real world. On the other hand, it remains to be seen if any location-based services providers will deploy cryptographic systems in the market.

2.4. Privacy preservation against attacks

Usually, there are four types of attacks faced by continuous queries: the homogeneity attack, the query sampling attack, the replay attack, and the query tracking attack.

2.4.1. Homogeneity attack

Homogeneity attack (Bettini et al., 2007) is launched in case of the lack of diversity among users in the anonymizing set with respect to locations or queries. To counter the query homogeneity attack, Liu et al. (2009) defined the query *l*-diversity to ensure that all queries in the same anonymizing set are different enough so that a query is hard to be linked to a certain user.

2.4.2. Query sampling attack

To defend against the query sampling attack (Chow and Mokbel, 2007; Pan et al., 2012), Chow and Mokbel (2007) introduced the concept of *k*-sharing region, i.e., a cloaked region not only covers at least *k* users but is treated as the cloaked region by at least *k* users.

2.4.3. Replay attack

Wang and Liu (2009) presented the replay attack model, which estimates the likelihood of some locations being a user's actual positions by rerunning the anonymizing algorithm. It should be noted that the resilience to replay attack is correlated with the anonymizing algorithm itself.

2.4.4. Query tracking attack

Query tracking attack (Chow and Mokbel, 2007) identifies potential query issuers by linking consecutive snapshots. To defend against this attack, Chow and Mokbel (2007) utilized the memorization property, which memorizes users in a cloaked region of a continuous query at the time when the query is initiated.

In our work, typical attacks faced by continuous queries are resisted by tailoring the cloaking algorithm for road networks. Features of moving trend, velocity difference, and distance difference are considered. To facilitate the cloaking process, we construct a hierarchical structure of road networks and corresponding maintenance strategies are provided in case of road networks update.

3. System model

In this section, we formulate the privacy-preserving problem first, then introduce the privacy profile and the corresponding privacy-preserving mechanism. Finally, we show the implementation strategies.

3.1. Problem formulation

We define the underlying road network and the privacy problem to be addressed.

3.1.1. The underlying road network

We consider a space restricted by the underlying road network, which is represented by a weighted directed graph $G=(V,E)$,

where the vertex set $V=\{v_0, v_1, \ldots, v_N\}$ stands for road junctions, and the edge set $E=\{(v_i, v_j)|\, v_i, v_j \in V\}$ represents road segments connecting two junctions v_i and v_j. The listed order $v_i v_j$ indicates the direction of the road segment from v_i to v_j . Note that in our model, the direction of a user's movement is preserved. When no confusion occurs and to simplify, we do not explicitly mention the directions of the underlying road network in figures appearing in subsequent sections.

We use $d(v)$ to denote the degree of a vertex v in V. Specifically, a vertex with $d(v)=1$ is called *end vertex*, an *intermediate vertex* has $d(v)=2$, and an *intersection vertices* has $d(v)\geq 3$. Each edge e in E is associated with a non-negative weight $w(e)$, which represents the cost of an edge from one vertex to the other. The cost can be the travel distance, trip time or toll of a corresponding road. In our system, we weigh edges with the travel distance and order vertices in the road network, based on which, we define the moving direction towards the vertex with a larger number as positive, otherwise, it is negative. In our work, all mobile users are assumed to reside in edges.

Combined with the underlying road network, the trace of a user u issuing a continuous query is a sequence of connected edges: $T_u=\{(v_{s1}, v_{e1}), (v_{s2}, v_{e2}), \ldots, (v_{sn}, v_{en})\}$, where v_{si} and v_{ei} denote the start node and the end node, respectively, of the ith edge passed by u, and $v_{ei}=v_{s(i+1)}$.

3.1.2. Problem settings

In continuous location-based services, a query has three statuses: (1) *New*: A newly initiated query is called *new query*. (2) *Active*: A query that was created before but not terminated yet is *active query*. (3) *Expired*: A query reaching its expiring time and being terminated is called *expired query*. For a new query, a mobile user sends it to a LBS provider in the form of $<u, l, T_{init}, T_{exp}, Con>$, where u is the identifier of the user, l is the user's current location (latitude and longitude), T_{init} represents the query initiating time, T_{exp} is the query expiring time, and *Con* is query text, such as "Inform me of the nearest petrol station in the next 30 min". Once it turns active, the user only needs to update his location l with his identifier u and sends it to the LBS provider, because the provider will preserve *Con* until T_{exp}. During the query lifetime, the LBS provider provides service for the user by answering the query periodically (e.g., every 30 s) with the updated locations.

Both locations and query contents are exposed to the LBS providers, which may be untrustworthy. Considering some LBS need users' exact locations for service provision, we try to preserve the query privacy of a user. In our system, we introduce a trusted third party to cloak a user with others into a cloaked user set S_u. Correspondingly, the cloaked segment set S_{sg} contains the segments that users in S_u reside in, and the queries sent by users in S_u form the cloaked query set Q.

3.1.3. Attack model

In order to explain our methods accurately, we establish the attack model against whom the preservation is placed. Generally, two characteristics are used to represent an attacker: *background knowledge* and *attacks*. We firstly specify an attacker's background knowledge and then we demonstrate the attacks he performs in order to steal privacy and harm individuals.

We assume that users' location and answers to queries can reveal nothing about query content, that is, the query issued by a user is unknown even if his locations are leaked. The background knowledge *BK* of an attacker about user is assumed to know:

1. u's exact locations during his query lifetime.
2. u's cloaked user set S_u and corresponding query set Q for each snapshot.
3. The privacy-preserving algorithms.

Given the employed privacy-preserving algorithms, the users' exact locations and cloaked user and query sets that are generated by the privacy-preserving algorithms, the attacker can run four typical attacks, which are most frequently and particularly implemented against continuous queries, namely, the query sampling attack (Chow and Mokbel, 2007; Pan et al., 2012), query tracking attack (Chow and Mokbel, 2007), replay attack (Wang and Liu, 2009), and homogeneity attack (Bettini et al., 2007). All of them aim to find-out associations between users and queries.

Homogeneity attack is due to lack of diversity, we use query entropy to measure the diversity of a cloaked query set, which will be explained in Section 3.2. With the diversity, homogeneity attack can be naturally resisted by the privacy-preserving algorithms, therefore, we only consider the other three attacks. Generally, as the attacker obtains the background knowledge, he tries to infer some private information of interest about the users' query content, such as linking a user's exact location to a specific query and having access to the query content. Nevertheless, users are cloaked together in a region with the form of a cloaked user set and queries sent by them are grouped into a cloaked query set. Therefore, the problem of linking a user's exact location to his actual query is probabilistic. The output of the attack can be a probability distribution on the possible categories of attacks. Hence, we define *linkability* to quantify the vulnerability of our framework under the three typical attacks.

Definition 1 (*Linkability*). The linkability of query q to user u under *BK*, denoted as $\text{link}[u \leftarrow q|\, BK]$, is the probability that an attacker can infer q is issued by u among users in the cloaked user set S_u.

Query sampling attack (Chow and Mokbel, 2007): Query sampling attack means that when the distribution of users' locations is not uniform, cloaked user sets overlap with each other. Therefore, some users can be cloaked into two or more sets, which increases the probability of linking the query to the query issuer.

Query sampling attack can be formalized as follows: suppose there are three users u_1, u_2, u_3, respectively, issuing queries q_1, q_2, q_3. S_{u1} containing u_1 and u_2 is the cloaked user set of u_1 while S_{u2} containing u_2 and u_3 is that of u_2. An attacker can infer $\text{link}[u_1 \leftarrow q_1|\, BK]=1$, as u_1 only belongs to S_{u1}.

Query tracking attack (Chow and Mokbel, 2007): In continuous queries, users continuously report their locations to LBS providers. Query tracking attack can link consecutive time snapshots together to identify a query issuer, although he is cloaked with other users.

Suppose user u issues query q. At time t_1, he is cloaked into a user set S_{u1} and the corresponding query set is Q_1. Hence, the *Linkability* is $\text{link}[u \leftarrow q|\, BK]=\frac{1}{|Q_1|}$. With time passing by, more cloaked sets are generated, denoted as S_{ui} and Q_i for time t_i. As the query issuer must be in all the cloaked user sets, an attacker links the sets and the *Linkability* is changed to $\text{link}[u \leftarrow q|\, BK]=\frac{1}{|Q_1 \cap Q_2 \cap \cdots \cap Q_n|}$.

Replay attack (Wang and Liu, 2009): In the replay attack, we assume that an attacker has full knowledge regarding the cloaking algorithm. By rerunning the cloaking algorithm with an element in the cloaked user set assumed to be the query issuer, the attacker estimates the likelihood of the user to generate the cloaked set.

An attacker replays the cloaking process as follows: for each user $u_i \in S_u$, (1) re-runs the cloaking algorithm by taking u_i as the query issuer of query q to generate a cloaked set S'_{ui}, with $|S_u|=|S'_{ui}|$; (2) calculates the probability of u_i to issue q, that is, $\text{Prob}[S_u|\, u_i, BK]=\frac{|S_u \cap S'_{ui}|}{|S_u|}$; and (3) select u_i with the largest probability value as the query issuer. The *linkability* is $\text{link}[u \leftarrow q|\, BK]=\frac{\text{Prob}[S_u|\, u, BK]}{\sum_{i=0}^{n} \text{Prob}[S_u|\, u_i, BK]} \times \frac{1}{|Q|}$.

In this paper, we aim to prevent linking a continuous location-based query to a specific user, i.e., low *linkability*, under the query sampling attack, query tracking attack, and replay attack.

3.2. Privacy profile

As mentioned above, S_u, S_{sg} and Q, respectively, signify the cloaked user set, cloaked segment set, and corresponding query set. A road segment is a sequence of edges $(v_0v_1, v_1v_2, \ldots, v_{m-1}v_m)$, among which only v_0 and v_m are intersection vertex or end vertex. The generated S_u, S_{sg} and Q should satisfy a user's personalized privacy requirements defined in his privacy profile in the form of $(k_{local}, k_{global}, l_{local}, l_{global}, L_{max}, Dis_{max})$. They define privacy requirements mainly from four aspects: k-anonymity, l-diversity, maximum length and maximum distance.

3.2.1. k-anonymity

A query obeys k-anonymity (Chow and Mokbel, 2007) if it could be issued by any of k users. In our system, a query issuer is cloaked with at least k-1 other indistinguishable users to achieve k-anonymity.

k_{local} and k_{global} are requirements of k-anonymity. k_{local} ensures that the user is cloaked with at least $k_{local}-1$ other users at each snapshot. As for k_{global}, it indicates that the number of common users in intersection of the cloaked set for consecutive snapshots in a continuous query is at least k_{global}. The query tracking attack would fail as the query issuer is still indistinguishable from $k_{global}-1$ other users even if an attacker links all the cloaked sets. For a continuous query composed of n snapshots, we maintain that

$$|S_{ui}| \geq k_{local}$$
$$|S_{u1} \cap S_{u2} \cap \cdots \cap S_{un}| \geq k_{global}$$

where $S_{ui}(i = 1, 2, \ldots, n)$ is the cloaked user set of the ith snapshot.

3.2.2. Query l-diversity

For a cloaked query set Q, given an integer l, it satisfies query l-diversity if the query entropy of this set is equal to or greater than $\log(l)$.

Similar to yellow pages companies categorizing different businesses, we classify queries into different categories according to the Point of Interests (*POIs*), such as hospitals and restaurants. For example, a user issuing a query "Report me the nearest petrol station every 5 min in the next 30 min" seems to be interested in the petrol station, hence, the query pertains to the petrol station category. The set of categories is denoted as $C = \{c_1, c_2, \ldots, c_n\}$. Suppose that the query categories are already known, and the accurate query contents cannot be inferred from these categories. For a specific query q pertaining to category c_i, the query entropy H is defined as

$$H = -\sum p_i \log p_i$$

where p_i is the percentage of queries pertaining to c_i in Q:

$$p_i = \frac{|\{q \mid q \in Q,\ q.c = c_i\}|}{|Q|}$$

Query l-diversity is introduced to resist the homogeneity attack (Bettini et al., 2007). Similar to the k-anonymity restriction, we have l_{local}-diversity and l_{global}-diversity for each single cloaked query set and the intersection of all the sets. So we have

$$H(Q(S_{ui})) \geq \log l_{local}$$
$$H(Q(S_{u1} \cap S_{u2} \cap \ldots \cap S_{un})) \geq \log l_{global}$$

where $Q(S_{ui})$ is the set of cloaked queries issued by users in S_{ui}, $H(\cdot)$ is the entropy function.

3.2.3. Maximum length

A query fulfills the maximum length restriction when the total length of road segments in the cloaked segment set S_{sg} is not larger than the pre-defined maximum value.

In our system, it restricts the total weight of segments in S_{sg} to L_{max}, that is,

$$L(S_{sg}) \leq L_{max}$$

where $L(\cdot)$ is the total length of edges in S_{sg}.

L_{max} is introduced to limit the expansion of the cloaked set, which may raise computation and communication costs with more candidate results generated. The maximum length requirement is especially important in a coarse area where the population density is particularly low, because it needs a large cloaked segment set to satisfy k-anonymity. While k-anonymity within the maximum length restriction is violated, we can generate dummy queries consistent with the query context (Pingley et al., 2011). Hence, user's privacy is preserved as the attacker cannot tell the real one from dummies. Generating dummies is another topic in location-based queries privacy preservation, which is not the focus of this paper.

3.2.4. Maximum distance

A query satisfies the maximum distance requirement only if the distance between the query issuer and any of other cloaked users is less than the pre-defined maximum distance, denoted as Dis_{max} in our framework. Then for each user $u_i \in S_u$, it holds

$$Dis(u, u_i) \leq Dis_{max}$$

where u is the query issuer, $Dis(u, u_i)$ is the length of the shortest path from u to u_i. It plays an important role especially when users in an area have a high probability of requesting a certain category of queries.

3.3. Privacy-preserving mechanism

First of all, we present the basic concepts in our privacy-preserving mechanism. Then, we show the qualification for users to be cloaked.

3.3.1. Snet and Snet hierarchy

In the road network, a portion conceptually covered by a cluster (or community) can represent a potentially cloaked segment set. Inspired by privacy-preserving techniques partitioning the spatial domain into cells in Euclidean space, we construct sub-graphs of the road network recursively bottom-up. Each sub-graph is named as an *Snet*, which is the basic cloaking unit in our system.

Definition 2 (*Snet*). For a given road network graph $G = (V, E)$, an *Snet* is a sub-graph of G, which is denoted as $S_n = (V_s, B_s, E_s)$, where V_s, B_s, and E_s respectively denotes vertex set, border vertex set, and edge set in S_n, besides:

1. $E_s \subseteq E$.
2. $V_s = \{v \mid (v, v') \in E_s \vee (v', v) \in E_s\}$, where (v, v') is the edge linking v and v'.
3. $B_s = V_s \cap \{v \mid (v, v') \in E' \vee (v', v) \in E'\}$, where $E' = E - E_s$.

Figure 2 shows two *Snets* (in the dashed frame) of the road network graph G (in the solid frame). For the left *Snet*, the corresponding vertex set V_s is $\{v_1, v_2, v_3\}$, the edge set E_s contains (v_1, v_3) and (v_2, v_3), and the border vertex set B_s is $\{v_3\}$ because the edge (v_3, v_5) does not belong to the edge set E_s. Similarly, for the right *Snet*, the corresponding vertex set V'_s is $\{v_3, v_4, v_5\}$, the edge set E'_s contains (v_3, v_5) and (v_4, v_5), and the border vertex set B'_s is

$\{v_3, v_5\}$ because the edges (v_1, v_3), (v_2, v_3), and (v_5, v_7) are not in the edge set E'_s. In addition, because the two *Snets* share the border vertex v_3, they are called neighboring *Snets*.

We build the underlying road network into an *Snet hierarchy* by constructing *Snets* in a bottom-up manner, where *Snets* at upper levels are formed by *Snets* at lower levels. For simplicity, We limit that an *Snet* is composed of two sub-*Snets* at most. At each level, the road network is viewed as a graph of interconnected *Snets*. Specifically, each *Snet* at level 0 represents an original segment in the road network. There is only one *Snet* at the top-level *ht*, which covers the entire road network.

While constructing an *Snet*, $S_n(h+1,\cdot)$ at level $h+1$ with two sub-*Snets*, $S_n(h,i)$, $1 \le i \le 2$ at level h, where $S_n(h+1,\cdot) = (V_s(h+1,\cdot), B_s(h+1,\cdot)), E_s(h+1,\cdot)$, the following three conditions must be held:

1. Edges of *Snets* at level h are disjoint, i.e., $\forall i\ \forall j,\ i \ne j \rightarrow E_s(h,i) \bigcap E_s(h,j) = \varnothing$.
2. Edges in an *Snet* at level h only connect vertices in the same *Snet*, i.e., $\forall m \forall j,\ m \ne j, (v_m, v_j) \in E_s(h,i) \rightarrow v_m \in (V_s(h,i) \bigcup B_s(h,i)) \wedge v_j \in (V_s(h,i) \bigcup B_s(h,i))$.
3. For a *Snet* at level $h+1$ to be constructed, $S_n(h+1,\cdot) = (V_s(h+1,\cdot), B_s(h+1,\cdot), E_s(h+1,\cdot))$, where $V_s(h+1,\cdot)$ and $E_s(h+1,\cdot)$ are unions of the corresponding sets in the sub-*Snets* and $B_s(h+1,\cdot)$ is the union of the corresponding sets of the sub-*Snets*' border vertices, that is,
   - $V_s(h+1,\cdot) = \bigcup_{1 \le i \le 2} V_s(h,i)$;
   - $E_s(h+1,\cdot) = \bigcup_{1 \le i \le 2} E_s(h,i)$;
   - $B_s(h+1,\cdot) = \bigcup_{1 \le i \le 2} B_s(h,i) - \{v \mid v \in \bigcup_{1 \le i \le 2} B_s(h,i) \vee [(v,v') \in E_s(h+1,\cdot) \vee (v',v) \in E_s(h+1,\cdot)]\}$.

Definition 3 (*Transition probability*). The transition probability from edge i to j means the probability that users on edge i will move to j. It can be pre-computed by counting the times that users in i transfer into j according to history traces. Transition probability is calculated as the count of the transitions from edge i to edge j is divided by the total number of transition from edge i to other edges.

Algorithm 1. Building the *Snet* hierarchy.

Input $G=(V, E)$
Output $S_n(0,\cdot), S_n(1,\cdot) \ldots S_n(ht,\cdot)$.
1: $h \leftarrow 0$, $e \in E$ is denoted by $S_n(0,j)$, $V_{inter} \leftarrow \varnothing$, $E_{con} \leftarrow \varnothing$, $flag(e) = 0$
2: **while** $h < ht$ **do**
3: **for** $v \in V$ **do**
4: **if** $d(v) \ge 3$ **then**
5: $V_{inter} \leftarrow V_{inter} \bigcup \{v\}$
6: **end if**
7: **end for**
8: **for** $v_{inter} \in V_{inter}$ **do**
9: $E_{con} \leftarrow E_{con} \bigcup \{(\cdot, v_{inter})\} \bigcup \{(v_{inter}, \cdot)\}$
10: **if** $(\exists e_{con} \in E_{con}) \& (flag(e_{con}) = 0) \& (d(v') = 1 \| d(v') = 2)$, $(v_{inter}, v') = e_{con} \| (v', v_{inter}) = e_{con}$ **then**
11: initiation edge $e_{init} \leftarrow e_{con}$
12: **else**
13: $e_{init} \leftarrow e_{largest}$, where $(e_{largest} \in E_{con}) \& flow(E_{con})$ is the largest
14: **end if**
15: $S_n(h+1,j) \leftarrow$ merge e_{init} with highest transition possibility edge $e_{tranmax}$
16: **if** length $L(S_n(h+1,j)) \le 2^h \times$ Dis(E) **then**
17: $flag(e_{tranmax}) = 1, flag(e_{init}) = 1$
18: **else**
19: $flag(e_{init}) = 1$
20: **end if**
21: **end for**
22: **for** $e' \in E$
23: **if** $flag(e') = 0$ **then**
24: $S_n(h+1,j) \leftarrow \{e'\}$
25: **end if**
26: **end for**
27: each $S_n(h+1,j)$ is denoted by an edge e_{new}, $E \leftarrow \{e_{new}\}$
28: $h \leftarrow h+1$
29: **end while**
30: return $S_n(0,\cdot), S_n(1,\cdot) \ldots S_n(ht,\cdot)$

Generally, vertices with larger degrees play more important roles in road networks. Hence, we select intersection vertices and corresponding edges to initiate the construction process. Among all the neighboring edges, we give priority to those connected to an intermediate vertex or an end vertex. In case there is neither an intermediate nor an end vertex connected, we select the edge carrying the largest historical user flow as the initial edge. Then the edge that users are most likely to transfer into from the initial edge (i.e., the highest transition possibility) is selected to form an *Snet*. The formed *Snet* is further denoted by an edge at a higher level, which connects with neighboring *Snets* through the common border vertices. We recursively perform the *Snet* construction steps until the underlying road network is merged into one *Snet* at top-level *ht*. For an edge that is not merged with others, it assembles itself as an *Snet* at a higher level. To balance the privacy

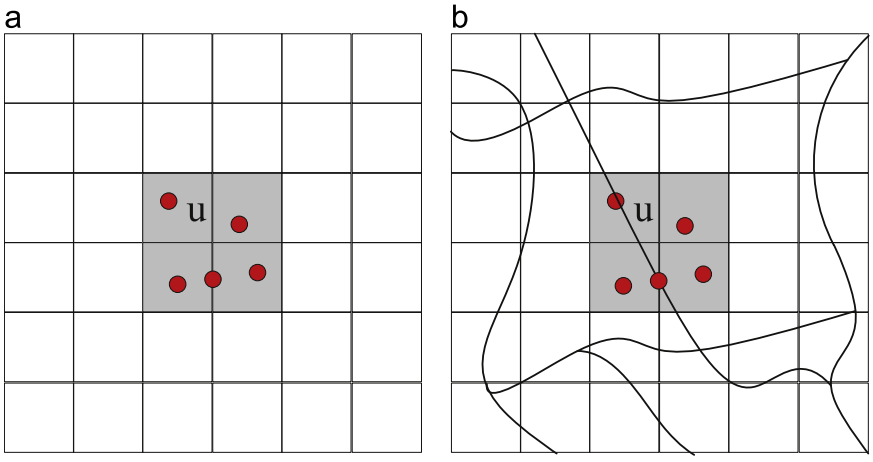

Fig. 1. Euclidean space and road networks. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

preservation and system costs, we restrict the maximum total length of edges in an *Snet* to the value L_{max}. Suppose the average edge length of the underlying road network is Dis(E), we use $2^h \times$ Dis(E) as the maximum length limitation for *Snets* at level h, because there are at most 2^h edges in an *Snet* at level h. The detailed process of building the *Snet* hierarchy is shown in Algorithm 1.

Figure 3 shows an example of building the *Snet* hierarchy for the road network in Fig 2. We use an edge to denote an *Snet*, the *Snets* encircled in the lower level means they will be constructed into a parent *Snet* in the upper level. The blue vertex v_5 is elaborately selected for the *Snet* construction process. The arrow indicates the transition direction from edge (v_4, v_5) to edge (v_5, v_3).

In a special case, each edge at level 0 (the raw underlying road network) constructs an *Snet*. For example, (v_1, v_3) denotes *Snet* $S_n(0,1)$. $S_n(1,1)$ represents an *Snet* formed by (v_1, v_3) and (v_2, v_3) at level 0. For the *Snet* construction, intersection vertex v_5 is selected. We pick (v_5, v_4) as the initiation edge, because it connects an end vertex v_4. We merge it with (v_3, v_5) which has the highest transition probability from (v_5, v_4) among neighboring edges (edges sharing a common border vertex). Thus, we get *Snet* $S_n(1,2)$ at level 1. Similarly, *Snets* $S_n(1,2)$ and $S_n(1,3)$ are represented by edges at level 1, and their neighboring *Snets* $S_n(1,1)$, $S_n(1,4)$, and $S_n(1,5)$ are connected though vertices v_3 and v_7. The *Snet* hierarchy construction process continues until the entire road network is merged into a single *Snet* $S_n(4,1)$ at the fourth level.

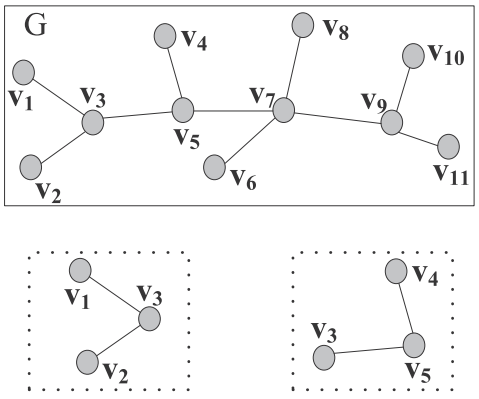

Fig. 2. An example of *Snet*.

3.3.2. Cloaking qualifications

Based on the *Snet* hierarchy, our system generates cloaked sets for users meeting predefined privacy profiles. To satisfy the k_{local} and l_{local} requirements, we expand the cloaked set in a bottom-up manner from *Snets* at level 0. The expansion process is terminated when the L_{max} requirement is violated. Simultaneously, a user with a distance to the query issuer longer than Dis_{max} will be kicked out. As for k_{global} and l_{global} requirements, we try to maintain users staying in the same *Snet* to remain in the same set in the long run.

There are three features that affect which *Snet* the user will enter and when he will enter it in the future: transfer behavior, velocity, and distance to a border vertex. We use *moving trend* to describe the transfer behavior that a user enters a certain *Snet* after leaving the previous one. Users moving into the same *Snet* in the future have the same moving trend. We treat users' velocities as vectors composed of moving directions and speed magnitudes, *velocity difference* is used to measure the velocity variations between users. Users with low *velocity difference* are more likely to stay close in the future. Similarly, we use *distance difference* to show users' difference in distance to the border vertex that they will pass through. Users with low *distance difference* are prone to stay in the same *Snet*. Thus, while selecting candidate users for cloaking, we prefer those with similar *moving trend*, low *velocity difference*, and low *distance difference* compared to the query issuer.

Moving trend: Users' *moving trend* can be modeled as *Markov* Chain on a set of neighboring *Snets* of the current *Snet* S_n. Let P^{S_n} be the transition matrix of S_n, the element $p_{ij}^{S_n}$, $i = 1 \ldots m$, $j = 1 \ldots w$ of P^{S_n} is the transition probability of users from edge i of S_n to *Snet* j, where m is the edge number of S_n, n is the amount of neighboring *Snets* of S_n.

Figure 4 shows an example of transition matrix. Let $P_n^S(1,2)$ be the transition matrix of *Snet* $S_n(1,2)$ with edges (v_3, v_5) and (v_4, v_5). Supposing the first row of $P_n^S(1,2)$ denoted as $p_{1,\cdot}^{S_n(1,2)}$ represents the

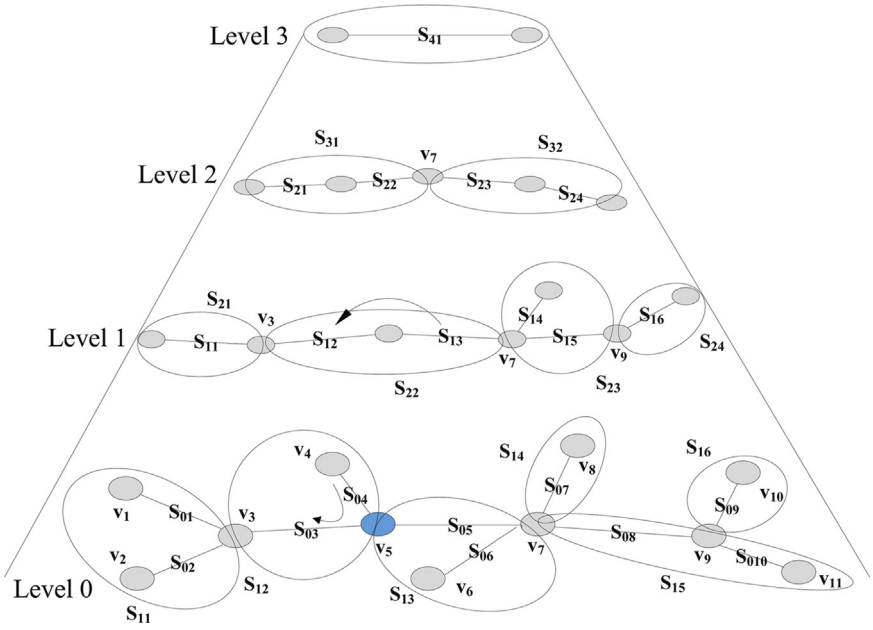

Fig. 3. An example of *Snet* hierarchy. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

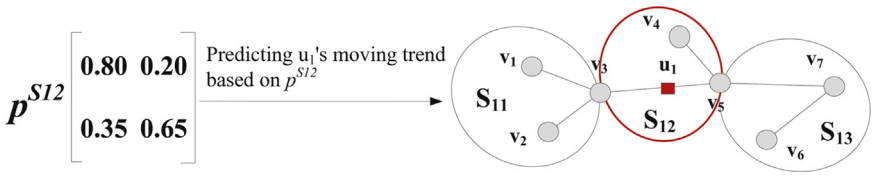

Fig. 4. An example of transition matrix. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

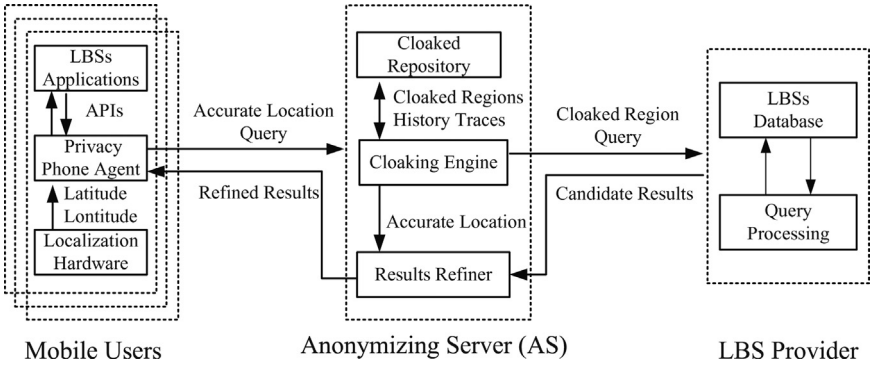

Fig. 5. System architecture.

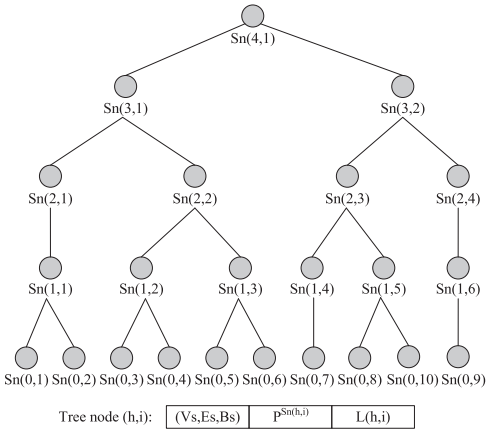

Fig. 6. The storage scheme.

transition probability of (v_3, v_5) to neighboring *Snets* $S_n(1,1)$ and $S_n(1,3)$ respectively, thus correspondingly, $p_{2,\cdot}^{S_n(1,2)}$ is the transition probability from (v_4, v_5) to $S_n(1,1)$ and $S_n(1,3)$. User u_1 denoted by a red rectangular is moving along edge (v_5, v_3) in $S_n(1,2)$. Obviously, u_1 has a higher probability to enter $S_n(1,1)$ after leaving $S_n(1,2)$.

Velocity difference: The difference between two users' velocities should consider both the moving direction and the magnitude. However, only users having the same moving trend need to check the velocity difference, those with opposite directions will be filtered out. Thus, we take the magnitude into account while calculating the velocity difference. The velocity difference VL_{diff} between users u_i and u_j is defined as

$$VL_{\text{diff}}(u_i, u_j) = ||vl_i| - |vl_j||$$

where $|vl_i|$ and $|vl_j|$ are the velocity magnitude of u_i and u_j.

Users qualified to be cloaked together should follow the velocity difference restriction ζ:

$$VL_{\text{diff}}(u_i, u_j) \le \zeta$$

Distance difference: The road network distance $d(u, v)$ between user u and vertex v is defined as the sum of edge weight along the shortest path from u to v. In our system, u will pass through v while entering the predicted *Snet*. v is a common border node of the two neighboring *Snet*. While there are more than one common border nodes, v is the one nearest to u. For u_i and u_j, they may enter the same *Snet* through different vertices, denoted as v_i and v_j. Then the distance difference between users to the vertex they may pass through is calculated as

$$D_{\text{diff}}(u_i, u_j) = |d(u_i, v_i) - d(u_j, v_j)|$$

Thus, users qualified to be cloaked together should further satisfy the distance difference restriction θ in the equation below:

$$D_{\text{diff}}(u_i, u_j) \le \theta$$

3.4. Framework implementation

In this section, we show the system architecture of our privacy-preserving framework. Then the storage scheme of the *Snet* hierarchy is discussed.

3.4.1. System architecture

Figure 5 shows the system architecture, it consists of three components: mobile users, the trusted Anonymizing Server (AS), and the LBS provider. A mobile user sends a query through the privacy phone agent to the AS with his privacy profile in the form of $< u, l, p, T_q, T_{\text{exp}}, Con >$, where $u, l, T_q, T_{\text{exp}}$ and Con mean the same with those in Section 3.1.2, p denotes the user's defined privacy profile $< k_{\text{local}}, k_{\text{global}}, l_{\text{local}}, l_{\text{global}}, L_{\max}, \text{Dis}_{\max} >$ discussed in Section 3.2. l is the user's location obtained by the positioning device. When the AS receives a query from a mobile user, the cloaking engine generates cloaked sets with the cloaking algorithms presented in Section 4. The AS sends queries issued by the cloaked users to the LBS provider, which returns the candidate results to the AS which follows the users and keeps track of their locations. The results refiner improves the results based on the user's accurate locations and forward the refined results to the privacy phone agent. The phone agent further transfers the result

to LBS applications. For a query in *Active* status, the mobile user periodically updates his locations until it turns *Expired*.

3.4.2. Snet storage scheme

Because the *Snet* at level $h+1$ is formed by two *Snets* at level h at most, we use the binary tree T to store the *Snet* hierarchy, which maintains the parent–child relationship of *Snets* at each level. Each *Snet* consisting of (V_s, E_s, B_s) is kept as a node. For each node, we store the transition matrix of the *Snet* discussed in Section 3.3.2. The total length of segments in the *Snet* is precomputed and stored in the node as well.

Figure 6 illustrates the storage structure of the *Snet* hierarchy in Fig. 3. Let the ith *Snet* at level h be $S_n(h,i)$, $L(S_n(h,i))$ be the total length of edges in $S_n(h,i)$, P^{S_n} be the transition matrix of $S_n(h,i)$. For our cloaking algorithms, the cloaked users are in the same tree node. In other words, for a query issuer residing in *Snet* $S_n(0,j)$, the users cloaked with him will be those in $S_n(0,j)$ or in one of its ancestors in the binary tree T.

4. Cloaking algorithms

We present two types of privacy-preserving algorithms based on the *Snet* hierarchy. The first one consisting of Algorithms 2 and 3 is designed for a single user, the other composed of Algorithms 4 and 5 is for a batch of users. Recalling that in the *Snet* construction process, an edge is merged with the edge that has the highest transition probability. Hence, our algorithms treat *Snet* as the basic cloaking unit and retrieves the binary tree storing the *Snet* hierarchy in the bottom-up manner. According to users' cloaking qualifications discussed in Section 3.3.2, our algorithms first retrieve the *Snet* at level 0, then select users with similar moving trend, satisfying the velocity difference and distance difference restrictions predefined by our system. The candidate sets are formed by the selected users, after which, users' privacy profiles will be checked. If the candidate set cannot fulfill users' privacy profiles, the algorithms search its parent *Snet*. These steps continues until users' privacy profiles are satisfied, or reach the top-level of the *Snet* hierarchy. If no qualified cloaked set is generated, corresponding queries will be terminated. As all queries sent to the LBS providers have to pass through the AS, the queries will be cut from the LBS providers if they are dropped by the AS. Hence, users' privacy can be preserved.

Compared with the first type of algorithms for a single user, the second type aims to improve the efficiency of our privacy-preserving framework. It generates a cloaked set for a batch of users simultaneously to decrease the cloaking time. In addition, the user's privacy is enhanced because the query sampling attack can be resisted by sharing the cloaked set among users.

4.1. Algorithms for a single user

We present algorithms for a single user in this section. When a sequence of users arrives, Algorithm 2 finds qualified users in an *Snet* at a certain level to form the candidate cloaked set for each user. Algorithm 3 generates the cloaked sets for a single user.

Algorithm 2. Selecting qualified users.

Input query $q<u,l,p,T_q,T_{exp},Con>$, *Snet* $S_n(h,i)$ and edge e of user u, binary tree T.
Output candidate cloaked set S_1.
1: predicting moving trend $S_n(h,j)$ of u, $S_e \leftarrow \varnothing$, $S_1 \leftarrow \varnothing$
2: **for** (e_i in $S_n(h,i)$)&($e_i \neq e$) **do**
3: **if** moving trend of users in $e_i = S_n(h,j)$ **then**
4: $S_e \leftarrow S_e \bigcup \{e\}$

Table 1
Example of privacy profile.

User	Query	Query category	k_{local}	l_{local}	k_{global}	l_{global}
A	q_A	c_1	2	2	2	2
B	q_B	c_2	3	2	2	2
C	q_C	c_1	3	3	3	2
D	q_D	c_3	3	3	3	2
E	q_E	c_3	2	2	2	2

Table 4
Experiment parameters.

Parameters	Values	Default	Unit
Number of users	2000	2000	–
Number of snapshots	50	50	–
Local privacy (k_{local}, l_{local})	2–5, 3–6, 4–7, 5–8, 6–9	2–5	/
Distance limit	1, 2, 3, 4, 5, 6	5	km

Table 2
Example of cloaked user set and query set.

User	S_1	Q_1	S_2	Q_2	S_3	Q_3
A	$\{A,E\}$	$\{q_1,q_3\}$	$\{A,E\}$	$\{q_1,q_3\}$	$\{A,E\}$	$\{q_1,q_3\}$
B	$\{A,B,D\}$	$\{q_1,q_2,q_3\}$	$\{A,B,D\}$	$\{q_1,q_2,q_3\}$	$\{A,B,E\}$	$\{q_1,q_2,q_3\}$
C	$\{B,C,E\}$	$\{q_1,q_2,q_3\}$	$\{B,C,E\}$	$\{q_1,q_2,q_3\}$	$\{B,C,E\}$	$\{q_1,q_2,q_3\}$
D	$\{A,B,D\}$	$\{q_1,q_2,q_3\}$	$\{A,B,D\}$	$\{q_1,q_2,q_3\}$	$\{A,B,D\}$	$\{q_1,q_2,q_3\}$
E	$\{A,E\}$	$\{q_1,q_3\}$	$\{A,E\}$	$\{q_1,q_3\}$	$\{A,E\}$	$\{q_1,q_3\}$

Table 3
Example of cloaked user set and query set.

User	S_1	Q_1	S_2	Q_2	S_3	Q_3
A	$\{A,E\}$	$\{q_1,q_3\}$	$\{A,E\}$	$\{q_1,q_3\}$	$\{A,E\}$	$\{q_1,q_3\}$
B	$\{B,C,D\}$	$\{q_1,q_2,q_3\}$	$\{B,C,D\}$	$\{q_1,q_2,q_3\}$	$\{B,C,D\}$	$\{q_1,q_2,q_3\}$
C	$\{B,C,D\}$	$\{q_1,q_2,q_3\}$	$\{B,C,D\}$	$\{q_1,q_2,q_3\}$	$\{B,C,D\}$	$\{q_1,q_2,q_3\}$
D	$\{B,C,D\}$	$\{q_1,q_2,q_3\}$	$\{B,C,D\}$	$\{q_1,q_2,q_3\}$	$\{B,C,D\}$	$\{q_1,q_2,q_3\}$
E	$\{A,E\}$	$\{q_1,q_3\}$	$\{A,E\}$	$\{q_1,q_3\}$	$\{A,E\}$	$\{q_1,q_3\}$

5: **end if**
6: **end for**
7: **for** u_i residing in S_e **do**
8: **if** $(VL_{diff}(u, u_i) \le \zeta)\&(D_{diff}(u, u_i) \le \theta)$ **then**
9: $S_1 \leftarrow S_1 \bigcup \{u_i\}$
10: **end if**
11: **end for**
12: return S_1

When selecting qualified users at each level, Algorithm 2 first predicts the *Snet* that user u will move into while leaving the original one (step 1). Then it predicts the moving trend of users staying in the same *Snet* with u, and picks out those having the same moving trend with u (steps 2–6). Among all the users picked out, only those satisfying the velocity difference and distance difference restrictions will be selected as qualified cloaked users (steps 7–11).

Algorithm 3. Cloaking for a single user.

Input query $q < u, l, p, T_q, T_{exp}, Con >$, binary tree T.
Output cloaked set S_i of the ith snapshot, $1 \le i \le m$ (totally m snapshots).
1: **if** q is *New* **then**
2: map l on edge e, find $S_n(0, j)$ containing e in T
3: $C_{temp} = S_n(0, j)$, $h \leftarrow 0$, $S_1 \leftarrow \varnothing$
4: **while** $(|S_1| < k_{local}) \| (H(Q(S_1)) < logl_{local})$ **do**
5: $S_1 \leftarrow$ *Selecting Qualified Users* (q, C_{temp}, e, T)
6: $C_{temp} \leftarrow C_{temp}$'s parent node, $h \leftarrow h+1$
7: **end while**
8: **if** $L(C_{temp}) > L_{max}$ **then**
9: suppress the query
10: **end if**
11: return S_1
12: **else**
13: **for** $u_j \in (S_1 \bigcup S_2 \bigcup \cdots \bigcup S_{i-1})$ **do**
14: **if** Distance$(u, u_j) \le Dis_{max}$ **then**
15: $S_i = S_i \bigcup \{u_j\}$
16: **end if**
17: **end for**
18: **if** $(|S_i \bigcap S_1 \bigcap S_2 \bigcap \cdots \bigcap S_{i-1}| \ge k_{global})\&$ $(H(Q(S_i \bigcap S_1 \bigcap S_2 \bigcap \cdots \bigcap S_{i-1})) \ge \log(l_{global}))$ **then**
19: return S_i
20: **else**
21: suppress the query
22: **end if**
23: **end if**

As is shown in Algorithm 3, user u sends a query in the form of $< u, l, p, T_q, T_{exp}, Con >$. When receiving a query q in *New* status from user u, the AS maps l into the road network. Algorithm 3 treats *S(0, j)* that containing edge e where u resides as the initial cloaked set (step 2). *S(0, j)* is a leaf node in the binary tree T. The algorithm traverses the binary tree until k-anonymity and l-diversity requirements of u are fulfilled (steps 3–7). When the total length of edges in the cloaked set is larger than L_{max}, the algorithm stops (steps 8–10). If the query is in *Active* status, the algorithm checks common users in the previous *i-1* cloaked sets, adds those satisfying the distance restriction into the cloaked set (steps 13–17). k_{global} and l_{global}-diversity requirements are also checked (steps 18–22).

4.2. Algorithms for a batch of users

As multiple queries may arrive at the AS simultaneously, cloaking for each respective user is inefficient. Following the reciprocity principle, we propose an optimized algorithms composed of Algorithms 4 and 5, which generates cloaked sets for a batch of users at one time. On the other hand, algorithms for a single user are vulnerable to the query sampling attack. Therefore, we introduce the k-sharing method to resist this attack. In our system, the generated cloaked set is shared by all users in it. Thus, while generating the cloaked set qualified for all users in it, we ensure the strictest privacy requirements of all the users cloaked, i.e., $k_{localmax}$, $k_{globalmax}$, $l_{localmax}$, $l_{globalmax}$, L_{maxmin}, Dis_{maxmin}. $k_{localmax}$, $k_{globalmax}$, $l_{localmax}$ and $l_{globalmax}$ indicate the maximum value of k_{local}, k_{global}, l_{local}, l_{global} defined by the users in the candidate cloaked set, L_{maxmin} and Dis_{maxmin} stand for the minimum value of users defined L_{max} and Dis_{max} restrictions. When the strictest privacy requirements can't be satisfied, the user requesting for such requirements should be kicked out from the candidate cloaked set. Similar to the first type of algorithms for a single user, the second type of algorithms for a batch of users includes two parts: Algorithm 4 selects the qualified users, and Algorithm 5 generates the cloaked set for a batch of users.

Algorithm 4. Selecting qualified users in batches.

Input query set $\{q < u, l, p, T_q, T_{exp}, Con >\}$, *Snet* set $\{S_n\}$, and binary tree T.
Output a set of candidate cloaked sets $\{S_i\}$.
1: **for** $S_n(h, i) \in \{S_n\}$ **do**
2: $\{CL\} \leftarrow \varnothing$, $\{S_i\} \leftarrow \varnothing$ 3: **if** $(u \in U(S_n(h, i)))\&(L_{max}(u) < L(S_n(h, i)))$ **then**
4: $U(S_n(h, i)) \leftarrow U(S_n(h, i)) - \{u\}$
5: **end if**
6: **for** $u \in U(S_n(h, i))$ **do**
7: predicting its moving trend $S_n(h, j)$
8: **end for**
9: $\{CL\} \leftarrow \{CL\}\{CL | CL \leftarrow \{$users with the same $S_n(h, j)\}\}$
10: **end for**
11: **for** $CL_i \in \{CL\}$ **do**
12: put the users in CL_i satisfying $((VL_{diff}(u_i, u_j) \le \zeta)$ and $(D_{diff}(u_i, u_j) \le \theta))$ into S_i
13: **end for**
14: return $\{S_i\}$

For the uncloaked users, Algorithm 4 removes user u violating its $L_{max}(u)$ restriction (steps 2–4). Then it predicts the moving trend of all the users and clustered them with the same moving trend together (steps 5–8). For each clustered user set, those following the velocity difference and distance difference restrictions are selected as the qualified users (steps 10–12).

Algorithm 5 maps all users into the road network, finds corresponding *Snets* in the binary tree (steps 3–6). It repeatedly selects qualified users until the $k_{localmax}$ and $l_{localmax}$ values of all users in the *Snet* are fulfilled (steps 9–12). When there are no candidates in the cloaked set or the traverse step goes beyond the top level, we suppress users with $k_{localmax}$ or $l_{localmax}$ requirements, and traverse back for one level (steps 13–20). The cloaking set is shared by all users residing in it. For the following snapshots, for each cloaked set, we check the maximum distance limit of all users residing in the previous *i-1* cloaked sets (step 28–32), after which, the k_{global} and l_{global} requirements will also be checked (steps 33–35).

Algorithm 5. Cloaking for a batch of users.

Input query set $\{q<u,l,p,T_q,T_{exp},Con>\}$, binary tree T.
Output set of the cloaked set $\{S_i\}$ of the ith snapshot, $1\le i\le m$ (totally m snapshots), $S_{ij}\in\{S_i\}$

if $\{q\}$ is *New* **then**
  $C\leftarrow\varnothing$, $h\leftarrow 0$, $\{S_1\}\leftarrow\varnothing$
  for $q_i\in\{q\}$ **do**
    map q_i on edge e_i, find $S_n(0,j)$ containing e_i in T
    $C=C\bigcup S_n(0,j)$
  end for
  $C_{temp}\leftarrow C$, $\{S_1\}\leftarrow$ selecting qualified users in batches (q,C,T)
  for $S_{1i}\in\{S_1\}$ **do**
    while $(|S_{1i}|<k_{localmax})\|(H(Q(S_{1i}))<\log(l_{localmax}))$ **do**
      $S_{1-temp}\leftarrow$ selecting qualified users in batches (q,C_{temp},T)
      $S_{1i}\leftarrow S_{1i-temp}$, $S_n(h,j)\leftarrow S_n(h,j)$'s parent node, $h\leftarrow h+1$, $C_{temp}=\{S_n(h,j)\}$
    end while
    if $(S_n(h,j)=S_n(ht,j))\|(|S_{1i}|=\varnothing)$ **then**
      $S_n(h,j)\leftarrow S_n(h,j)$' child node
      while $(|S_{1i}|<k_{localmax})\|(H(Q(S_{1i}))<\log(l_{localmax}))$ **do**
        if $(u\in S_{1i})\&((k_{local}=k_{localmax})\|(l_{local}=l_{localmax}))$ **then**
          $S_{1i}\leftarrow S_{1i}-\{u\}$
        end if
      end while
    end if
    $\{S_1\}=\{S_1\}\bigcup S_{1i}$
  end for
  for $u\in(\{q\}-\{S_1\})$ **do**
    mark u as *New*
  end for
  return $\{S_1\}$
else
  for $u_i\in(S_{1j}\bigcup S_{2j}\bigcup\ldots\bigcup S_{(i-1)j})$ **do**
    if distance$(u,u_i)\le Dis_{maxmin}$ **then**
      $S_{ij}=S_{ij}\bigcup\{u_i\}$
    end if
  end for
  if $(|S_{ij}\bigcap S_{1j}\bigcap S_{2j}\bigcap\ldots\bigcap S_{(i-1)j}|\ge k_{globalmax})\&(H(Q(S_{ij}\ S_{1j}\ S_{2j}\ \ldots\ S_{(i-1)j}))\le\log(l_{globalmax}))$ **then**
    $\{S_i\}=\{S_i\}\bigcup S_{ij}$
  end if
  return $\{S_i\}$
end if

4.3. The framework maintenance

To maintain the longtime effectiveness of our framework, we discuss the maintenance strategy in the presence of updating users' history traces and changing road network structures.

4.3.1. Users' history traces update

The population density and the transition probability of each *Snet* are calculated based on users' history traces. For a new user moving from edge a to edge b, only the population of edges a and b, and the transition probability from a to b should be increased. Hence, we update the population density of edge a and edge b, and the transition matrix of the *Snet* containing edge a will also be updated. Because users' movements have localization properties, the population density stays stationary under the influence of users' history traces in the long run. Hence, we keep the *Snet* hierarchy structure unchanged.

4.3.2. Road network structure update

As rebuilding the *Snet* hierarchy leads to expensive cost, we prefer to incrementally update the *Snet* hierarchy when the road network structure changes. Because we directly use the existing map information, the update frequency of the *Snet* hierarchy is consistent with that of the map, which may be about every six to twelve months. The road network is modeled as a weighted directed graph, therefore, its structural changes are in two ways, i.e., change of edge lengths, and change of edge relations.

1. *Change of edge lengths*: When an edge length changes (e.g., travel distance, trip time, or toll cost increases/decreases), the total length of the *Snets* containing the edge should be updated.
2. *Change of edge relations*: When new roads are constructed or existing roads are closed, the network topology will be changed. These changes can be depicted as adding or deleting edges in the *Snet* hierarchy.
   Adding a new edge: A newly added edge (v,v') connects vertex v

and v'. Let the edge set connecting v be E_v, connecting v' be E'_v. If all edges in E_v and E'_v belong to the same *Snet* at level 1, edge (v, v') is merged with the *Snet*. Otherwise, (v, v') is added to the *Snet* having most neighboring edges. *Snets* at the parent level are updated recursively.

Deleting an existing edge: When removing an edge (v, v') from the road network, it affects the *Snets* containing (v, v'). As a result, we only update the *Snets* containing (v, v') by deleting it from their edge set.

5. Attack resilience analysis

In this section, we analyze the proposed algorithms' resilience to the query sampling attack, query tracking attack, and replay attack.

5.1. Algorithms for a single user

Replay attack: Under the replay attack (Section 3.1.3), an attacker repeatedly runs the algorithms to generate S'_i with u as the input. It can be seen that $S_i \bigcap S'_i$ contains u. The *linkability* satisfies $\frac{1}{|S|\times|Q|} \le \text{link}[u \leftarrow q|BK] < \frac{1}{(\frac{(|S|-1)}{|S|}+1)\times|Q|}$. Hence, an attacker can identify the query issuer with the maximum probability $\frac{1}{2}$ *iif* $|S \bigcap S'_i| = 1$ with regard to all S'_i. However, this is practically impossible. Users in the cloaked set share the same predicted moving trend, within the velocity difference and the distance difference restrictions, i.e., all the users in the cloaked set tend to stay in nearby road segments, that is, $|S \bigcap S'_i| \gg 1$.

Query sampling attack: Under the query sampling attack (Section 3.1.3), an attacker observes the cloaked sets samples $S_1, S_2, \ldots, S_i$ with the corresponding query sets $Q_1, Q_2, \ldots, Q_i$. Hence, the *linkability* can be calculated as $\text{link}[u \leftarrow q|BK] = \frac{1}{|Q_1 \bigcap Q_2 \bigcap \ldots Q_i|}$.

Query tracking attack: By considering the characteristic of the query tracking attack (Section 3.1.3), we introduce the principle of k_{global} and l_{global}. The number of common users should be at least k_{global} and the entropy of the common query set should not be less than $\log(l_{\text{global}})$. Consequently, the probability of identifying a specific user's query, i.e., the *linkability* $link[u \leftarrow q|BK]$, is $\frac{1}{|Q_1|}$ at least and $\frac{1}{l_{\text{global}}}$ at most.

In addition, an attacker can combine the replay attack, query sampling attack, and query tracking attack to infer a user's query content. As a result, the *linkability* changes to $\text{link}[u \leftarrow q|BK] = \frac{\text{link}[S|u,BK]}{\sum_{u_i \in S}\text{link}[S|u_i,BK]} \times \frac{1}{|\sum_{t=1}^{n}\sum_{i=1}^{m}\bigcap Q_t^i|}$, where Q_t^i represents the query set Q_i containing u at time t, n is the number of snapshots and m is the number of set containing u. Accordingly, the *linkability* to a specific user is $\text{link}[u \leftarrow q|BK] = \frac{1}{|\sum_{t=1}^{n}\sum_{i=1}^{m}\bigcap Q_t^i|}$

Example 1. We suppose that there are five users in the system. Their privacy profiles are partially listed in Table 1. The cloaked user set S_i and its corresponding query set Q_i of each user for snapshot i are shown in Table 2. In Table 2, q_j means a query pertaining to category c_j.

We suppose that an attacker observes a cloaked user set sample $\{A, E\}$ and the corresponding query set sample $\{q_1, q_3\}$, but the attacker does not know for whom the set is generated nor the relations between the queries and the users. Hence the attacker runs the algorithms respectively with A and E as the input, gets the same cloaked results. However, he still does not know the query issuer. The $\text{Prob}[S|A, BK]$ is calculated as

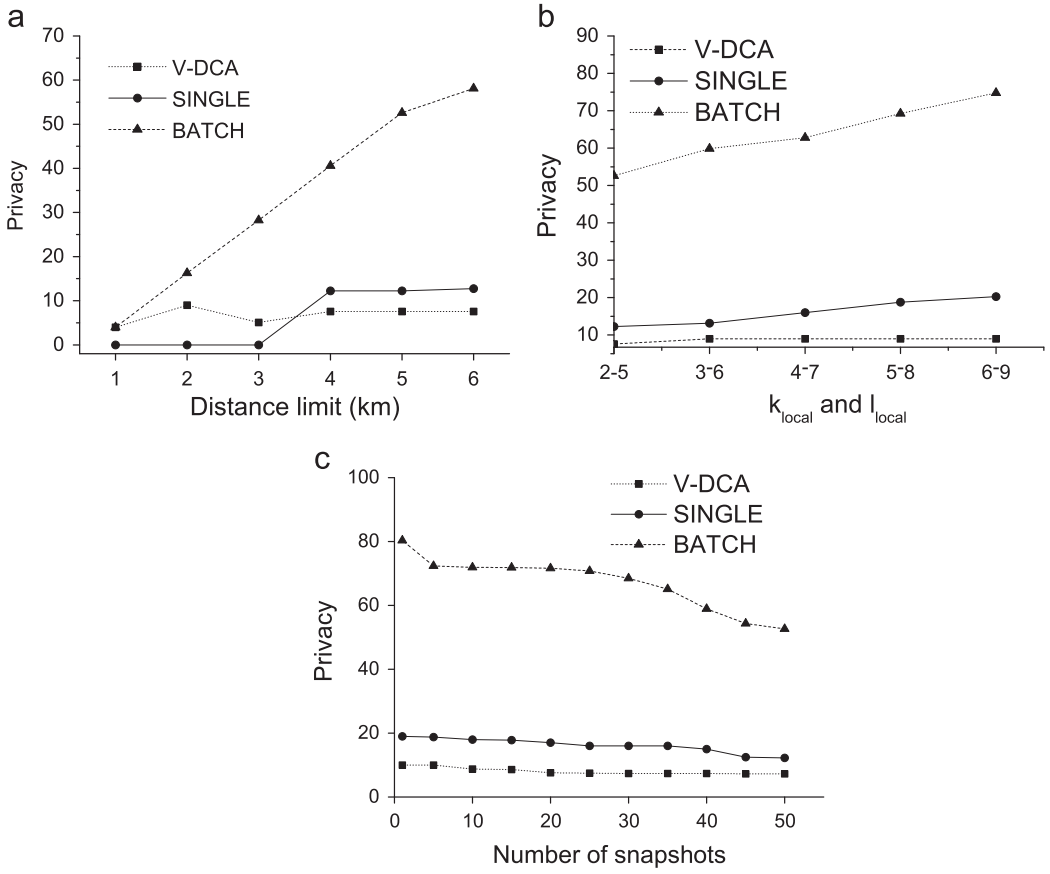

Fig. 7. Privacy-preserving ability evaluation.

$\frac{|\{A,E\}\cap\{A,E\}|}{|\{A,E\}|}=1$, so is $Prob[S|E,BK]$. Hence, the *linkability* of the replay attack is $\text{link}[A\leftarrow q_1|BK]=\frac{\text{Prob}[S|A,BK]}{\text{Prob}[S|A,BK]+\text{Prob}[S|E,BK]}\times\frac{1}{2}=\frac{1}{4}$. It is because the attacker cannot assure that A is the issuer and does not know $A's$ query is q_1. Under the query sampling attack, if the attacker observes that A is in $\{A,E\}$, $\{A,B,D\}$ at the first snapshot, he can infer $A's$ query must be in $\{q_1,q_3\}$ with the probability of 1/2. For the query tracking attack, if the attacker knows $B's$ cloaked user set and cloaked query set during snapshot 1, 2, and 3, he can infer $B's$ query with the probability of $1/|Q_1\cap Q_2\cap Q_3|=1/3$.

5.2. Algorithms for a batch of users

Because a cloaked set is generated for all users in it, the algorithms for a batch of users can effectively defend against the replay attack and query sampling attack. Therefore, we only analyze its resilience to the query tracking attack (Section 3.1.3). Similarly, the probability of identifying a specific user's query, i.e., the *linkability* $link[u\leftarrow q|BK]$ is $\frac{1}{l_{\text{global}}}$ at the most. The association between all users and queries has at least $\sum_{i=0}^{l_{\text{global}}}(-1)^i\binom{l_{\text{global}}}{i}(l_{\text{global}}-i)^{k_{\text{global}}}$ kinds of assignments. Hence, the probability

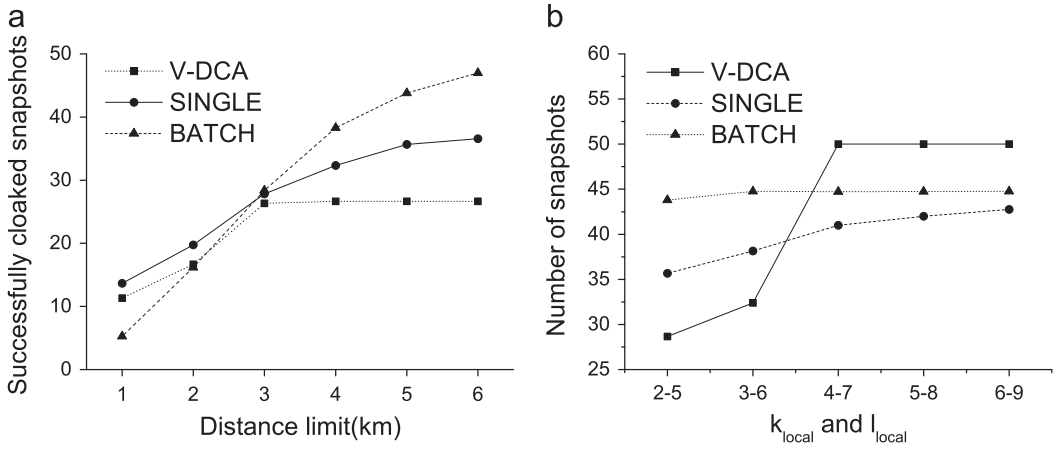

Fig. 8. Snapshots maintenance evaluation.

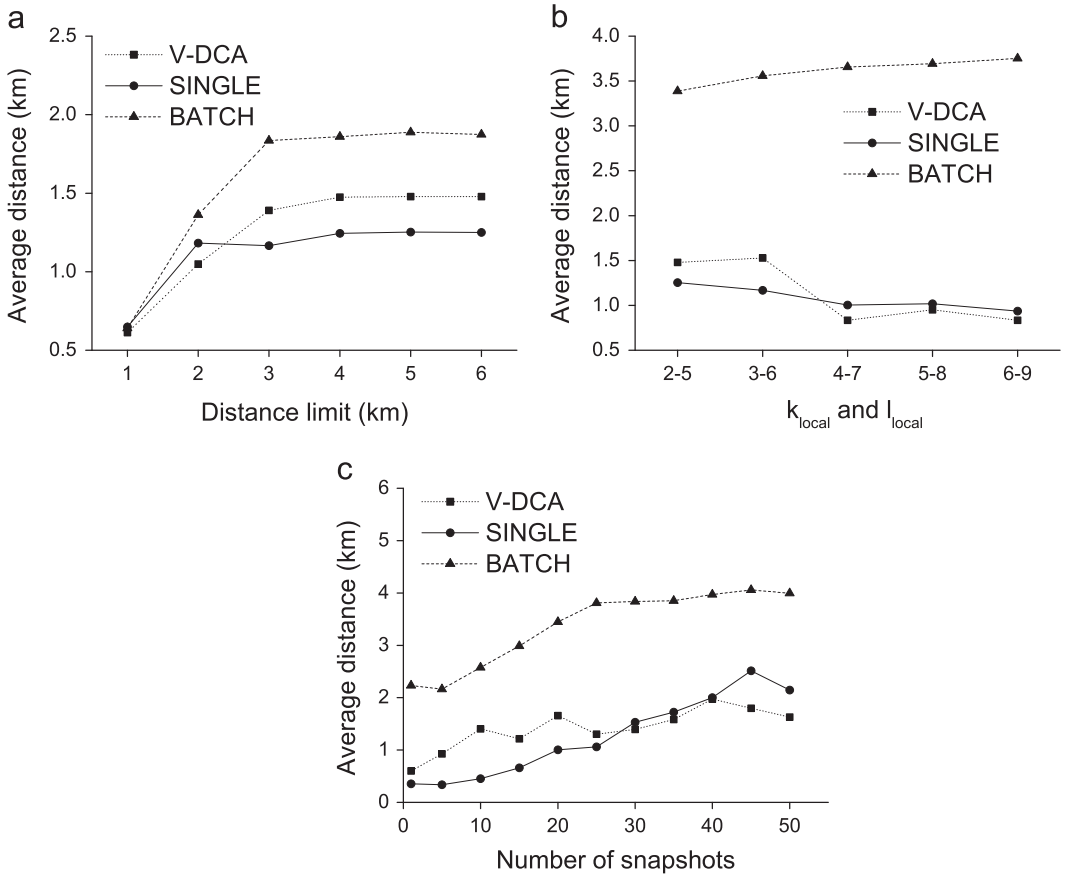

Fig. 9. Quality of service evaluation.

of an attacker successfully infers the user's query is

$$1/\sum_{i=0}^{l_{\text{global}}}(-1)^i\binom{l_{\text{global}}}{i}(l_{\text{global}}-i)^{k_{\text{global}}}$$

Example 2. Suppose there are five same users as is shown in Table 1. Their cloaked user set S_i and corresponding query set Q_i for snapshot i are listed in Table 3. If an attacker knows user $A's$ cloaked user set and query set during snapshot 1, 2 and 3, he can infer $A's$ query with probability of $1/|Q_1 \cap Q_2 \cap Q_3| = 1/2$.

6. Experiments and evaluations

In this section, we evaluate the effectiveness of our proposed algorithms. As there is no privacy-preserving approach for continuous queries in road networks according to our knowledge, we will do the comparison with another algorithm named *V-DCA* (Wang et al., 2012b) designed for Euclidean space. *V-DCA* is a continuous query privacy-preserving approach taking the velocity and acceleration features of users into consideration while cloaking. However, it does not take the underlying road network into consideration, let alone building the network hierarchy to facilitate the cloaking process. The evaluation criteria and metrics are presented, followed by the experiments setup description. Then, the evaluating results are discussed in detail.

6.1. Evaluation criteria and metrics

We evaluate the algorithms from three aspects: privacy-preserving ability, quality of service and performance. Cloaking algorithms generate cloaked sets for users meeting their privacy profiles. The predefined parameters, such as k_{global} and k_{local}, are users' privacy requirements. However, the effectiveness of the algorithms depends on the real achieved values of the cloaked set. Hence, we use the real archived values as the metrics to evaluate our framework. Correspondingly, we denote these values as K_{local}, K_{global}, L_{local}, L_{global}, *Len*, and Dis.

6.1.1. Privacy-preserving ability

For a continuous query, the privacy level depends on the common users and their queries of all the cloaked sets. We use *privacy* to measure the privacy level of a user in a continuous query, which can be computed as

$$\text{privacy} = K_{\text{global}} \times L_{\text{global}}$$

Higher *privacy* means a better privacy preservation. Similarly, we use the number of successfully cloaked snapshots n to measure the maintenance of algorithms. A better privacy-preserving method can provide service for a user longer, that is, larger n.

6.1.2. Quality of service

We evaluate the quality of service with the average distance of users in a cloaked segment set S_{sg} as query answers are more accurate within a smaller cloaked region. Hence, a smaller value of distance indicates better quality of service. While cloaking for a single user, the average distance $D_{\text{avg}}(u, S_{\text{sgi}})$ of the cloaking segment set S_{sgi} of the ith snapshot is calculated as

$$D_{\text{avg}}(u, S_{\text{sgi}}) = \frac{\sum_{i=1.to.K_{\text{local}}-1} D(u, u_i)}{K_{\text{local}} - 1}$$

where u is the query issuer, u_i represents user i cloaked with u in

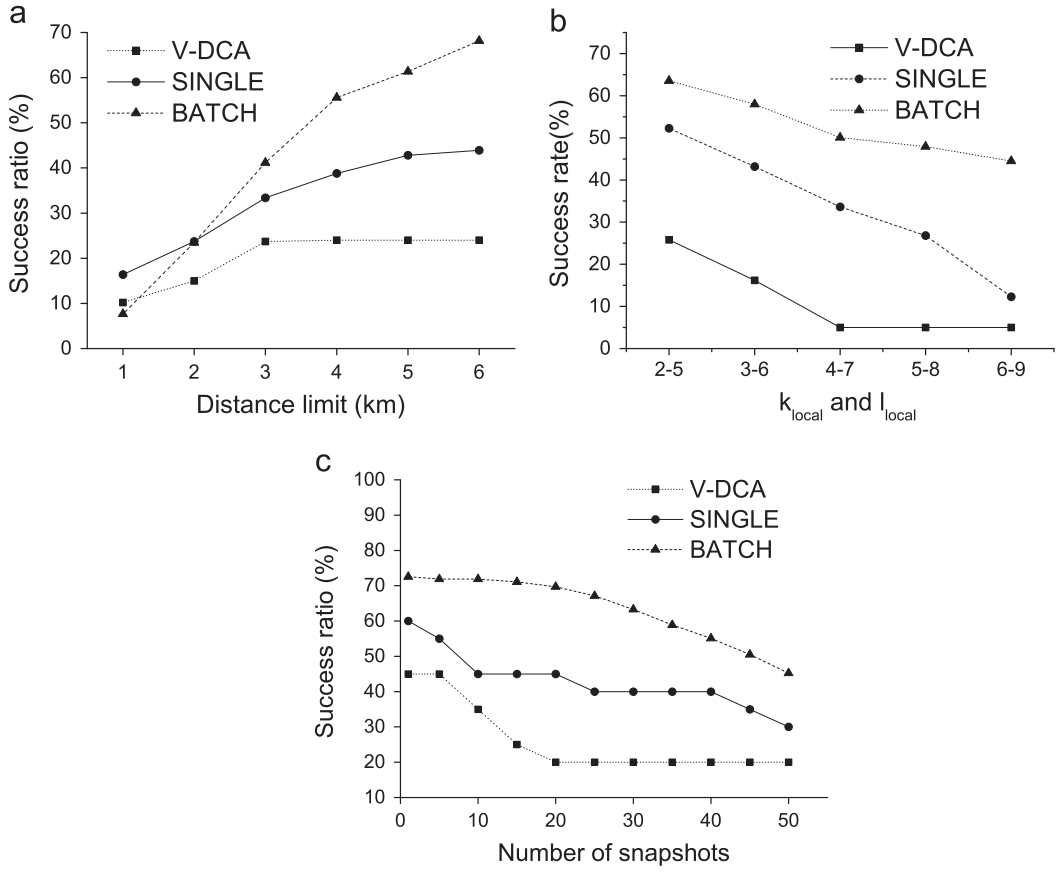

Fig. 10. Success ratio evaluation.

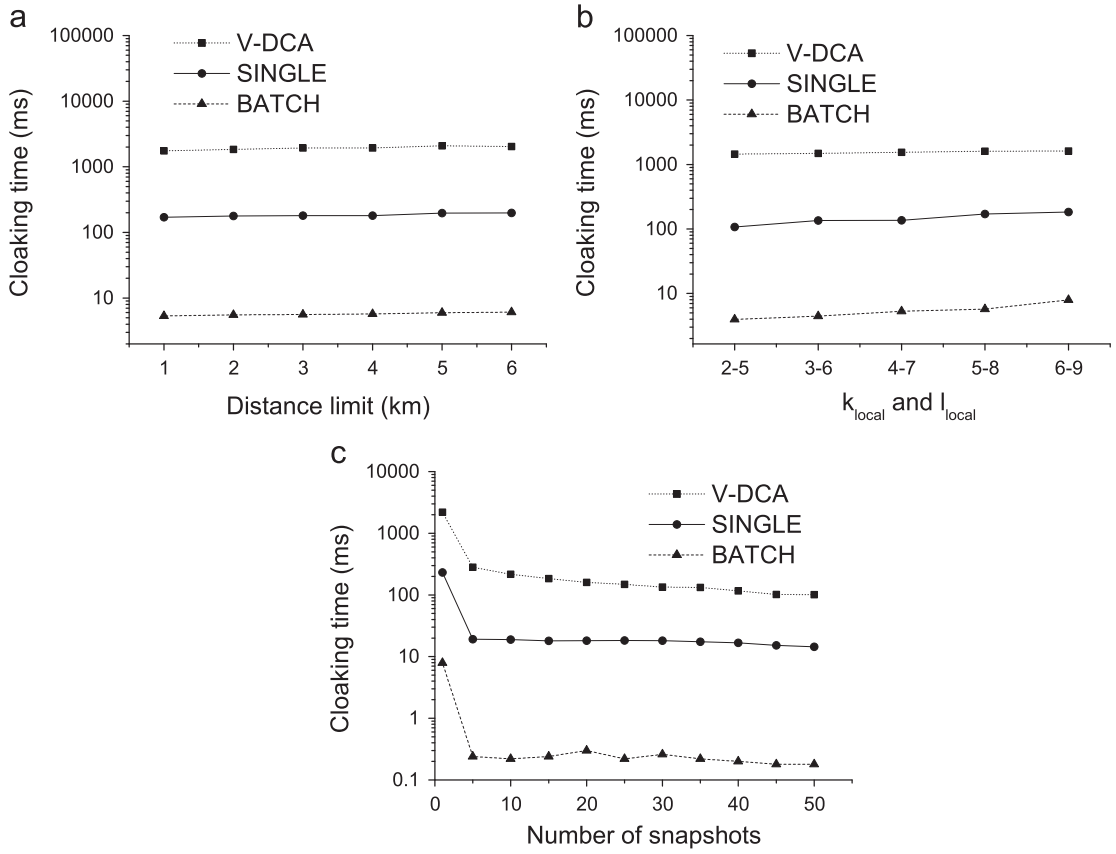

Fig. 11. Cloaking time evaluation.

S_{sgi}. The average distance $D_{avg}(u)$ of n consecutive snapshots is

$$D_{avg}(u) = \sum_{i=1.to.n} D_{avg}(u, S_{sgi})/n$$

Meanwhile, for a batch of users, the average distance

$$D_{avg}(U, S_{sgi}) = \frac{\sum_{j=1.to.K_{local}-1} D_{avg}(u_j, S_{sgi})}{K_{local}}$$

where U is the user set, u_i represents user i residing in U. The average distance of n consecutive snapshots is

$$D_{avg}(U) = \sum_{i=1.to.n} D_{avg}(U, S_{sgi})/n$$

6.1.3. Performance

We evaluate the performance of our framework from two aspects: cloaking success ratio and cloaking time.

1. *Cloaking success ratio*: It is the percentage of users that are successfully cloaked:

   $$SR = \frac{\sum |S|}{|U|}$$

   where S is the set of successfully cloaked users and U is the set of all the query issuers. A higher cloaking success ratio corresponds to a better performance.
2. *Cloaking time*: We use t_i to represent the cloaking time for snapshot i. For a well-performed privacy-preserving mechanism, the cloaking time should be short enough to achieve an enjoyable user experience.

6.2. Experiment setup

We use Thomas Brinkhoff Network-based Generator of Moving Objects (Brinkhoff, 2002) on the road map of Oldenburg. 2000 mobile users are generated moving along the road network with medium speed for 50 snapshots. Users' privacy requirements, such as k_{local} and l_{local}, are set randomly within a certain range. For example, the default range of k_{local} and l_{local} is 2–5. The maximum distance limit mentioned in Section 3.2 ranges from 1 km to 6 km. Parameters used in our experiment are listed in Table 4. The default values are used if they are not specifically described in the following experiments.

The simulating experiments are carried out using a PC with Dual Dore 2.13 GHz CPU, 4 GB RAM memory, and Windows 7 × 32 ultimate operating system. We implement the algorithms with C++. For all the graphs, *SINGLE* denotes the proposed cloaking algorithms for a single user, correspondingly, *Batch* is that for a batch of users. We repeatedly run each experiment for ten times and take the average values as the evaluation results. The standard deviation error bars are negligible.

6.3. Evaluation results

Figure 7 shows the privacy-preserving abilities of *V-DCA* and the two types of algorithms we proposed. It can be seen from Fig. 7 (a) that our algorithms perform better than *V-DCA* when the distance exceeds 4 km. Specifically, the privacy of *SINGLE* remains constant until the distance limit exceeds 4 km, while the privacy of *Batch* grows linearly and performs the best among them. Figure 7 (b) shows that the privacy of *V-DCA* and *SINGLE* stay nearly stable with the change of k_{local} and l_{local} (named as local privacy), while that of *Batch* keeps increasing and stays much higher than *V-DCA* and *SINGLE*. As shown in Fig. 7(c), over time, i.e., with the snapshot

number going up, the privacy of *BATCH* keeps the highest among the three algorithms.

The results of Fig. 7 indicate that *BATCH* provides the best privacy preservation among the three algorithms. Generally, *SINGLE* performs better than *V-DCA*.

Figure 8 shows that all the three algorithms provide privacy preservation for more snapshots with the growth of the distance limit or the increase of the local privacy value (i.e., k_{local} and l_{local}). In Fig. 8(a), *V-DCA* performs well with strict distance limit and maintains a relatively lesser number of snapshots when the distance limit exceeds 3 km. *SINGLE* performs better than *V-DCA* because of its consideration of users' moving trend and distance relations. *BATCH* is more easily influenced by the distance limit because it needs to balance the privacy requirements, users' velocities and distance among users in a cloaked region. It performs the best and can achieve nearly 50 snapshots with a looser distance limit. Figure 8(b) shows that local privacy can hardly affect the maintenance of *SINGLE* or *BATCH*. They perform pretty well even if local privacy is very low. *V-DCA* can successfully cloak for 50 snapshots when local privacy is larger than 4. This is because more users are cloaked for the first snapshot and are candidates to be chosen into the cloaked sets for the following snapshots. The results of Fig. 8 indicate that a looser distance limit and a higher local privacy is conducive to the maintenance of queries. However, the number of maintained snapshots grows slowly while distance limit and local privacy increase greatly.

We use average distance to measure the quality of service. As shown in Fig. 9, *BATCH* has larger average distance than the other two because it considers all users instead of the centered user in the cloaked region to provide an efficient cloaking function. We consider the moving trend and distance difference into the cloaking process, *SINGLE* can provide better quality of service than *V-DCA* in the long run. The three algorithms can effectively select users staying together in the following snapshots because of their consideration of users' movement features. In addition, all of them perform fairly steady with the change of distance limit, local privacy and snapshot numbers. Combining with Figs. 7 and 8, it can be told that our proposed algorithms have a good balance between privacy and quality of service.

Figure 10 evaluates the success ratio influenced by the distance limit and the local privacy. It is obvious that *SINGLE* and *BATCH* can obtain higher success ratio under the same distance limit and local privacy requirements than *V-DCA*, because they take the underlying road network properties into account. Furthermore, *BATCH* performs the best due to the validity of the cloaked set for all the users within the region. Our proposed algorithms can maintain much more users than *V-DCA* in the long run. Especially, *BATCH* can successfully anonymize for about 45 percent users even to the 50th snapshot.

Figure 11 shows the average cloaking time for each user with these three algorithms. It can be seen that *SINGLE* takes approximately one in ten time of that *V-DCA* takes, and *BATCH* only takes about a twentieth of the time that *SINGLE* takes except for the first snapshot. The main reason is that *SINGLE* and *BATCH* are based on the *Snet* hierarchy structure which improves the speed of retrieving users to be cloaked together. Furthermore, *BATCH* cloaks for a batch of users instead of for a single user at one time, as a result, it is more efficient.

From all the evaluated results, we can conclude that our proposed algorithms can achieve better privacy preservation than *V-DCA* while maintaining quality of service and improving the success ratio. Due to the initiation process of building the *Snet* hierarchy, the cloaking time can be decreased.

7. Conclusion

In this paper, we proposed a fast continuous LBS query privacy-preserving framework in road networks. As shown in the above statement, the framework considers the topological properties of the road network when providing privacy-preserving mechanisms for a single user and a batch of users. The analysis and experimental results indicate that our algorithms can resist typical attacks and preserve users' query privacy effectively in road networks.

References

Bamba B, Liu L, Pesti P, Wang T. Supporting Anonymous Location Queries in Mobile Environments with PrivacyGrid. In the 17th International Conference on World Wide Web (WWW), 2008 pp. 237-246.

Bao J, Chen H, Ku WS. Pros: a peer-to-peer system for location privacy protection on road networks. In: 17th ACM SIGSPATIAL international conference on advances in geographic information systems (GIS); 2009. p. 552–3.

Bettini C, Jajodia S, Pareschi L. Anonymity and diversity in LBS: a preliminary investigation. In: 5th IEEE international conference on pervasive computing and communications workshops (PERCOMW); 2007. p. 577–80.

Brinkhoff T. A framework for generating network-based moving objects. GeoInformatica 2002;6(2):153–80.

Chor B, Kushilevitz E, Goldreich O, Sudan M. Private information retrieval. J ACM 1998;45.

Chow C-Y, Mokbel MF. Enabling private continuous queries for revealed user locations. In: Proceedings of 10th international conference on advances in spatial and temporal databases (SSTD); 2007. p. 258–73.

Chow C-Y, Mokbel MF, Bao J, Liu. X. Query-aware location anonymization for road networks. Geoinformatica 2011;15(3):571–607.

Domingo-Ferrer J. Microaggregation for database and location privacy. In: 6th international conference on next generation information technologies and systems (NGITS); 2006. p. 106–16.

Durr F, Skvortsov P, Rothermel K. Position sharing for location privacy in non-trusted systems. In: 2011 IEEE international conference on pervasive computing and communications (PERCOM); 2011. p. 189–96.

Freudiger J, Shokri R, Hubaux J-P. On the optimal placement of mix zones. In: 9th international symposium on privacy enhancing technologies (PETS); 2009. p. 216–34.

Gedik B, Liu L. Protecting location privacy with personalized k-anonymity: architecture and algorithms. IEEE Trans Mob Comput 2008;7(1):1–18.

Ghinita G, Kalnis P, Khoshgozaran A, Shahabi C, Tan K-L. Private queries in location based services: anonymizers are not necessary. In: Proceedings of the ACM SIGMOD international conference on management data; 2008.

Gruteser M, Grunwald D. Anonymous usage of location-based services through spatial and temporal cloaking. In: 1st international conference on mobile systems, applications and services (MobiSys); 2003. p. 31–42.

Guha S, Jain M, Padmanabhan VN. Koi: a location-privacy platform for smartphone apps. In: Proceedings of the NSDI '12; 2012.

Huang Y, Vishwanathan R. Privacy preserving group nearest neighbor queries in location-based services using cryptographic techniques. In: Global telecommunications conference (GLOBECOM); 2010. p. 1–5.

Kainis P, Ghinita G, Mouratidis K, Papadias D. Preventing location-based identity inference in anonymous spatial queries. IEEE Trans Knowl Data Eng 2007;19 (12):1719–33.

Kolahdouzan M, Shahabi C. Voronoi-based K nearest neighbor search for spatial network databases. In: Thirtieth international conference on very large data bases, vol. 30 (VLDB); 2004. p. 840–51.

Ku WS, Zimmermann R, Peng WC, Shroff S. Privacy protected query processing on spatial networks. In: 2007 IEEE 23rd international conference on data engineering workshop (PDM); 2007. p. 215–20.

Li XY, Jung T. Search Me If you can: privacy-preserving location query service. In: Proceedings of the IEEE INFOCOM; 2013.

Liu FY, Hua KA, Cai Y. Query *l*-diversity in location-based services. In: 2009 tenth international conference on mobile data management: systems, services and middleware (MDM); 2009. p. 436–42.

Liu XX, Zhao H, Pan M, Yue H, Li X, Fang Y. Traffic-aware multiple mix zone placement for protecting location privacy. In: IEEE INFOCOM; 2012. p. 972–80.

Mokbel MF, Chow CY. Aref WG. The new casper: query processing for location services without compromising privacy. In: 32nd international conference on very large data bases (VLDB); 2006. p. 763–74.

Mouratidis K, Yiu ML. Anonymous query processing in road networks. IEEE Trans Knowl Data Eng 2010;22(1):2–15.

Narayanan A, Thiagarajan N, Lakhani M, Hamburg M, Boneh D. Location privacy via private proximity testing. In: Proceedings of the network distributed system security conference; 2011.

Olumofin F, Tysowski PK, Goldberg I, Hengartner U. Achieving efficient query privacy for location based services. In: 10th international conference on privacy enhancing technologies (PETS); 2010. p. 93–110.

Palanisamy B, Liu L. MobiMix: Protecting location privacy with mix-zones over road networks. In: 2011 IEEE 27th international conference on data engineering (ICDE); 2011. p. 494–505.

Pan X, Xu J, Meng X. Protecting location privacy against location-dependent attacks in mobile services. IEEE Trans Knowl Data Eng 2012;24(8):1506–19.

Papadias D, Zhang J, Mamoulis N, Tao Y. Query processing in spatial network databases. In: 29th international conference on very large data bases, vol. 29 (VLDB); 2003. p. 802–13.

Papadopoulos S, Bakiras S, Papadias D. Nearest neighbor search with strong location privacy. In: Proceedings of the VLDB endowment; 2010.
Pingley A, Zhang N, Fu XW, Choi H-A, Subramaniam S, Zhao W. Protection of query privacy for continuous location based services. IEEE INFOCOM 2011:1710–8.
Samarati P, Sweeney L. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical Report SRI-CSL-98-04. Computer Science Laboratory, SRI International; 1998.
Wang T, Liu L. Privacy-aware mobile services over road networks. In: Proceedings of the VLDB endowment, vol. 2, issue no. 1; 2009. p. 1042–53.
Wang Y, Xu DB, He X, Zhang C, Li F, Xu B. L2P2: location-aware location privacy protection for location-based services. In: IEEE INFOCOM; 2012. p. 1996–2004.
Wang Y, He HL, Peng J, Zhang TT, Li HZ. Privacy preserving for continuous query in location based services. In: IEEE 18th international conference on parallel and distributed systems (ICPADS); 2012. p. 213–20.
Wang Y, Peng J, He LP, Zhang TT, Li HZ. LBSs privacy preserving for continuous query based on semi-honest third parties. In: IEEE 31st international performance computing and communications conference (IPCCC); 2012. p. 384–91.
Zhong G, Goldberg I, Hengartner U. Louis Lester and Pierre: three protocols for location privacy. In: Proceedings of the 7th international conference on privacy enhancing technologies; 2007.