




# PrivSem: Protecting location privacy using semantic and differential privacy

Yanhui Li<sup>1,2</sup>  · Xin Cao<sup>2</sup> · Ye Yuan<sup>1</sup> · Guoren Wang<sup>1</sup>

Received: 3 April 2018 / Revised: 13 February 2019 / Accepted: 10 April 2019 /  
Published online: 27 April 2019  
© Springer Science+Business Media, LLC, part of Springer Nature 2019

## Abstract

In this paper, we address the problem of users' location privacy preservation on road networks. Most existing privacy preservation techniques rely on structure-based *spatial cloaking*, but pay little attention to locations' semantic information. Yet, the semantics may disclose sensitive information of mobile users. In addition, these studies ignore the location privacy requirements of other users, which is essential for location-based services (LBS). Thus, to tackle these problems, we propose *PrivSem*, a novel framework which integrates *location k-anonymity*, *segment l-semantic diversity*, and *differential privacy* to protect user location privacy from violation. In this framework, rather than using the original location data, we only access to the sanitized data according to differential privacy. Due to the nature of differential privacy which perturbs the real data with noise, it is particularly challenging to determine an effective *cloaked area*. Further, we investigate an error analysis model to ensure the effectiveness of the generated cloaked areas. Finally, through formal privacy analysis, we show that our proposed approach is effective in providing privacy guarantees. Extensive experimental evaluations on large real-world datasets are conducted to demonstrate the efficiency and effectiveness of *PrivSem*.

**Keywords** Location privacy · *l*-semantic diversity · Location *k*-anonymity · Differential privacy

## 1 Introduction

With the proliferation of mobile communication devices loaded with positioning capabilities, recent years have witnessed the explosive growth of location-based services (LBS) on road networks. Typical examples of these applications include location-based store finder,

---

✉ Yanhui Li  
lyhneu506822328@163.com

<sup>1</sup> Northeastern University, Shenyang, China

<sup>2</sup> The University of New South Wales, Sydney, Australia

road navigation, and location-aware advertisement. While offering great benefits and new business opportunities, the LBS also presents new threats—the intrusion of location privacy [2, 3, 38, 43, 55]. Adversaries can exploit user location information for such malicious purposes as spamming, stalking, and inferring health condition or alternative lifestyle habits.

Over the past years, many promising approaches have been proposed to preserve users’ location privacy. Most of them focus on location  $k$ -anonymity or location  $l$ -diversity, which blur the exact location of a mobile user into a cloaked area (CR for short). To compute a CR, location  $k$ -anonymity extends it until at least  $k-1$  other users are included, and location  $l$ -diversity extends until  $l-1$  different locations are included. In this sense, an exact location is mixed with other users ( $k$ -anonymity) or other locations ( $l$ -diversity), which makes it difficult for an adversary to learn valuable information.

Both techniques guarantee some degree of privacy. Nevertheless, they have the following two serious drawbacks. First, the generated CR could breach semantic information, which potentially endangers the individuals’ privacy. More concretely, a CR might only include semantically homogeneous locations even when it is perturbed with other users and locations, and hence an adversary would be able to infer semantic meanings from the CR. In other words, it is vulnerable to a *semantic homogeneity attack*. Second and more importantly, all previous solutions do not take the location privacy of other users in the system into consideration, while providing the privacy protection for a location-based query user. However, this process may pose a serious threat to other users’ privacy, as illustrated by the following example.

*Example 1* Figure 1 shows an example of a query snapshot which consists of 8 mobile users  $\{u_1, u_2, \dots, u_8\}$ . Consider a scenario, a patient  $u_1$  named Bob uses his GPS-enabled mobile phone from road  $e_7$  to find the nearest Italian restaurant. This simple query can be answered by an LBS in a publicly available web server (e.g., Google Maps). However, the LBS is not trusted, Alice can collaborate with the LBS to acquire the location of Bob and infer his health status. Thus, to prevent Bob’s location from leakage, instead of directly sending the query to the LBS, he uses an anonymizer, which is a trusted server. The location anonymizer perturbs his exact location according to his specified privacy requirement before forwarding this query to the LBS. The state-of-the-art approach used by the location anonymizer is based on *segment  $l$ -diversity* [7, 45], which cloaks Bob’s walking road with other nearby roads. The set,  $\{e_6, e_7, e_9\}$  may be a CR under location  $k$ -anonymity ( $k = 5$ ) and segment  $l$ -diversity ( $l = 3$ ). Unfortunately, all roads  $e_6, e_7$  and  $e_9$  have the homogeneous semantics,

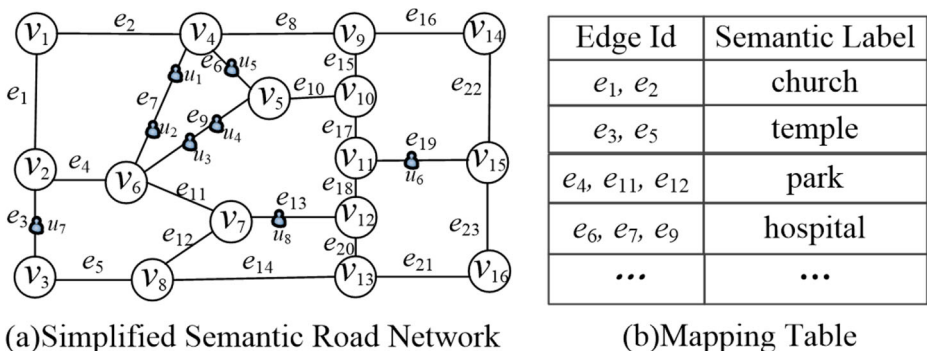


Figure 1 A snapshot of mobile users over a road network

namely hospital. Consequently, it is quite easy for an adversary to confidently derive that  $u_1$  is in the hospital. Hence, mobile users using  $\{e_6, e_7, e_9\}$  for the LBS query requests would be more likely linked to hospitals and could be suspected of having treatment. Alternatively, assume that an adversary has the knowledge of the locations of all users except  $u_3$ . By some technical means, he intercepts the generated CR  $\{e_6, e_7, e_9\}$ . Based on given  $k = 5$  and  $l = 3$ , he can definitely infer that  $u_3$  must be in this CR. However, this inference leaks  $u_3$ 's health status, which violates the privacy of  $u_3$ .

Several studies [10, 27, 53] have put significant effort to resist semantic homogeneity attacks over road networks, however, they have different critical limitations as stated in [29]. In addition, to our knowledge, no existing studies can effectively address the second drawback. In our early work [29], we have studied the problem of location privacy preservation of mobile users on road networks and proposed *l-semantic diversity* based EIRank algorithm. Unfortunately, the user identity is protected by using a pseudonym. In some scenarios, it is not adequate for providing identity protection [15]. Instead, in this paper, we propose to leverage **location  $k$ -anonymity** to protect user identity. The differences between this extended paper and our previous work [29] are described in Related Work.

Besides, ensuring the location privacy from leakage for other users is very important. While utilizing location  $k$ -anonymity to provide identity protection for query users, it requires to acquire the locations of other users. However, if the generated CRs suffer from the risk of location leakage, mobile users are reluctant to report their locations. In turn, it hinders the implementation of location  $k$ -anonymity, further impacts the effectiveness of identity protection. On the other hand, due to an increasing awareness of privacy risks, users might refrain from accessing the LBS, which would hinder the proliferation of these services. Hence, it is paramount to provide the location privacy for other users.

To tackle these problems, we propose *PrivSem*, a novel framework which integrates segment *l-semantic diversity*, location  $k$ -anonymity, and  $\epsilon$ -differential privacy, to protect users' location privacy over road networks. In this framework, the cloaking algorithm only has access to the location data sanitized according to *differential privacy (DP)*, rather than the original data. In practice, mobile users subscribe to a cellular service provider (CSP) that already has access to their locations (e.g., through cell tower triangulation). The CSP reaches an agreement with its subscribers on the terms and conditions of location disclosure. Therefore, it could release user locations to third party in a noisy form. However, using DP introduces the following two difficult challenges.

First, the generated CRs are determined by noisy location data, which requires sophisticated strategies to ensure the effectiveness of these CRs. To be specific, the determination of CRs is a serious dilemma: for the location  $k$ -anonymity, the generated CRs cannot be too small; yet, extending them would lead to more query results, and thus heavier system overhead in delivering these results. To create sanitized location data released at the CSP, we resort to the *Private Spatial Decomposition (PSD)* approach [8]. Typically, a PSD is a sanitized spatial index, where each node reports a noisy count of the users rooted at that node. To guarantee that the generated CR has a high success rate, we investigate an error analysis model that determines with high probability a spatial region that includes sufficient users. The other challenge is data sparsity in the spatial domain. In reality, compared to the total number of mobile users, the total number of roads could be very large. Consequently, the majority of PSD nodes have very low to zero count. The data sparsity poses great challenge for privacy preserving techniques since the perturbation noise is more likely to dominate the released count in presence of a small set of mobile users. To remedy this deficiency, we propose an *adaptive group mechanism* to resist the influence of noises on counting values.

In summary, we make the following contributions:

- For the first time, we detect the privacy leakage problem of other users in the context of location-based queries, and we present a framework that achieves differentially private guarantees.
- We have provided a cloaking technique that balances privacy requirements with system overhead in [29]. In this paper, we develop a series of optimization strategies to further improve its performance.
- We propose an error analysis model that quantifies the difference between noisy count of users in a CR and its real count, and we also introduce a search strategy that find appropriate PSD regions to ensure high success rate of the cloaking algorithm.
- Through formal privacy analysis, we prove that our proposed solution can provide required privacy protection. We conduct an extensive experimental study over real-world datasets. Experimental results demonstrate that our proposed techniques are effective and efficient.

The reminder of this paper is organized as follows. Section 2 presents some necessary preliminaries. Section 3 introduces the proposed privacy framework, whereas Sections 4 and 5 detail the proposed solution. The privacy guarantees of proposed solution are analyzed in Section 6, followed by the experimental evaluation in Section 7, and a survey of related work in Section 8. Finally, Section 9 concludes this paper.

## 2 Preliminaries

### 2.1 Road network

**Definition 1** (*Semantic Road Network*) A road network is modeled as an undirected graph  $G = (V, E, \xi)$  with a node set  $V$  and an edge set  $E$ , such that (i) a node  $v \in V$  represents a road intersection or a location (e.g., church); (ii) an edge  $e = (u, v) \in E$ , also referred as a **segment**, connects two nodes  $u$  and  $v$ ; and (iii)  $\xi$  represents a semantic function, i.e., for each edge  $e \in E$ ,  $\xi(e)$  is the sensitive semantic label of segment  $e$ .

*Example 2* (*Semantic Road Network*) Figure 1a gives an example of a semantic road network, in which each edge is associated with a semantic ID. The semantic labels corresponding to the IDs are shown in Figure 1b. Nodes  $v_4, v_5$  and  $v_6$  in Figure 1a are different buildings within the same hospital. Edges  $e_6, e_7$  and  $e_9$  connecting these three nodes would then have the same sensitive semantic label “hospital”. Thus, the area represented by the triangle ( $v_4, v_5$  and  $v_6$ ) would indicate the hospital.

### 2.2 Differential privacy

*Differential Privacy* (DP) has recently emerged as the state-of-the-art scheme for protecting individuals’ privacy. It is a semantic model which provides strong protection against realistic adversaries with auxiliary backgrounds. Informally, DP guarantees that the computation output of an algorithm is relatively insensitive to any change of one individual record, and thus, an adversary can learn limited information about any a specified individual. One important notion in DP is *neighboring datasets*. We say that two datasets  $D$  and  $D'$  to be neighboring, denoted by  $D \Theta D'$ , if  $|D| = |D'|$  and  $D$  and  $D'$  differ in only one location record.

**Definition 2** ( $\epsilon$ -Differential Privacy) A randomized algorithm  $\mathcal{A}$  is  $\epsilon$ -differential privacy if for any pair of neighboring datasets  $D$  and  $D'$ , and for any subset of output  $S \subseteq \text{Range}(\mathcal{A})$ ,

$$\Pr(\mathcal{A}(D) \in S) \leq e^\epsilon \Pr(\mathcal{A}(D') \in S) \quad (1)$$

The parameter  $\epsilon$  is called as the *privacy budget* which controls the level of privacy protection. A smaller  $\epsilon$  implies more restrictions imposed on the influence of a single user location, and hence gives more privacy to the individual. To achieve differential privacy, one principal technique is the *Laplace mechanism* [11], which injects random noise following Laplace distribution into the output. The amount of the noise depends on the *sensitivity* of query function  $f$ , formally defined as:

**Definition 3** (Sensitivity) The *sensitivity* of a query function  $f: D \rightarrow \mathbb{R}^d$ , is the maximum change caused by a single record. Formally,  $\Delta f = \max_{D \ominus D'} \|f(D) - f(D')\|_1$ , where  $\|\cdot\|_1$  is  $L_1$  norm.

**Theorem 1** (*Laplace Mechanism*) Given a function  $f: D \rightarrow \mathbb{R}^d$ , for any dataset  $D$ , a randomized algorithm  $\mathcal{A}_f$  that returns  $f(D) + \text{Lap}(\frac{\Delta f}{\epsilon})^d$  satisfies  $\epsilon$ -differential privacy, where  $\text{Lap}(\lambda)^d$  denotes a vector of  $d$  i.i.d. samples from the Laplace distribution  $\text{Lap}(\lambda)$ .

Two composition properties are widely used to ensure the overall privacy, known as sequential and parallel compositions.

**Theorem 2** (*Sequential Composition* [33]) Let  $\mathcal{A}_1, \dots, \mathcal{A}_m$  be  $m$  algorithms, each provides  $\epsilon_i$ -differential privacy. A sequential of algorithms  $\mathcal{A}_i(D)$  over the dataset  $D$  provides  $(\sum_i \epsilon_i)$ -differential privacy.

**Theorem 3** (*Parallel Composition* [33]) Let  $\mathcal{A}_1, \dots, \mathcal{A}_m$  be  $m$  algorithms, each provides  $\epsilon_i$ -differential privacy. Then, a sequential of  $\mathcal{A}_i(DS_i)$  over disjoint subsets  $DS_i$  of dataset  $D$  provides  $(\max_i \epsilon_i)$ -differential privacy.

In our approach, the noise injected may be from a sum of independent Laplace distributions instead of a single Laplace distribution. Thus, here we present two important results for sum of independent Laplace distributions.

**Lemma 1** (*sum of laplace distributions* [4]) Let  $Y = \sum_{i=1}^n \gamma_i$  be the sum of  $\gamma_1, \dots, \gamma_n$  independent Laplace random variables with zero mean and scale  $b_i$  and  $b_M = \max\{b_i\}$ . Let  $v \geq \sqrt{\sum_{i=1}^n b_i}$ , and  $0 < \lambda < \frac{2\sqrt{2}v^2}{b_M}$ . Then,  $\Pr(Y > \lambda) \leq \exp(-\frac{\lambda^2}{8v^2})$ .

**Corollary 1** (*measure concentration* [4]) Let  $Y, v, \{b_i\}_i$  and  $b_M$  be defined as in Lemma 1. Suppose  $0 < \delta \leq 1$  and  $v > \max\{\sqrt{\sum_{i=1}^n b_i}, b_M \sqrt{2 \ln \frac{2}{\delta}}\}$ . Then,  $\Pr[|Y| > v \sqrt{8 \ln \frac{2}{\delta}}] \leq \delta$ .

## 2.3 Private Spatial decompositions (PSD)

*Private Spatial Decompositions (PSD)* is introduced to release spatial datasets in a DP compliant manner [8]. Typically, a PSD is a sanitized spatial index transformed according to DP, where each node contains a noisy count of the data points rooted at that node. A variety of index types can be used as a basis for building PSD, such as k-d trees, grids or quad-trees.

Generally, the accuracy of PSD is extremely affected by the type of spatial structure and its parameters. The existing PSD can be divided into two categories: object-based PSD and space-based partitioning PSD. With object-based PSD, the split position for a node relies on the placement of user locations. This category includes structures such as k-d trees and R-trees. To protect privacy, split decisions also must be made according to DP, and a share of privacy budget is consumed in the process. Thus, the privacy budget must be split between building the index structure and reporting node counts. In theory, object-based PSD are more balanced, but they are not very robust. Only tiny changes of the PSD parameters can decrease the accuracy abruptly.

For space-based partitioning PSD, it performs splits of nodes only depending on the underlying structures (e.g., quad-trees, BSP-trees and grids), rather than real user location data. The privacy budget is entirely used to report the user count in each node. Generally, all nodes at the same level have non-overlapping domains, which yields a constant and low sensitivity of 2 per level. This is because, changing a single location in the dataset may affect at most two partitions in a level. The merit of space-based partitioning PSD is simple to construct, but can become unbalanced.

The accuracy of PSD also relies on the allocation of privacy budget. The best allocation for budget  $\epsilon$  across levels is *geometric allocation* [8], where higher levels receive less budget than leaf nodes. To ensure overall privacy, the *sequential composition property* is applied across nodes on the same root-to-leaf path, whereas *parallel composition property* is applied to disjoint paths in the hierarchy. In this paper, we adapt the space-based partitioning PSD to address the specific requirements of the LBS over road networks.

### 3 PrivSem: an overview

In this section, we first present the system architecture and the workflow for privacy-preserving location-based queries, and then introduce the privacy model and assumptions. Finally, we discuss design challenges and associated performance metrics.

#### 3.1 System architecture

The *PrivSem* adopts the classic centralized architecture for providing anonymous information delivery in the LBS, as sketched in Figure 2. In this architecture, the location anonymizer (LA) is a trusted entity which locates between mobile users and LBS providers. It consists of three tiers. The first tier is the user profile model which captures user personalized location privacy requirements. The second tier is comprised of the location cloaking components typically specialized in location anonymization service. The final tier is dedicated to the filter of candidate results. Besides, the communication between mobile users and the LA is via establishing an authenticated and encrypted connection.

More specifically, mobile users send their locations to a trusted cellular service provider (CSP) which periodically collects location updates and releases a PSD according to privacy budget  $\epsilon$  mutually agreed upon with the users. This CSP is either located in the LA or a dedicated server. Then, when the LA receives a query request with the exact location information from a query user (Step 1), it queries the PSD to determine a CR according to the user's privacy requirement, and relays it to the LBS provider (Step 2). Subsequently, the LBS provider computes the candidate results for received anonymous query, and forwards these produced results to the LA (Step 3). Finally, the LA extracts the exact answers from

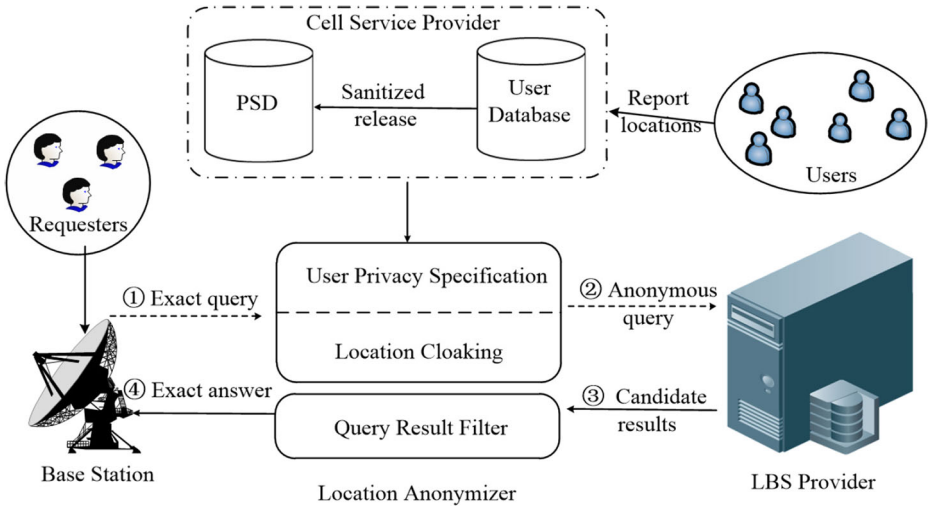


Figure 2 Overview of PrivSem

candidate results by properly filtering false hit information, and delivers them to the query user (Step 4).

### 3.2 Privacy model and assumptions

In *PrivSem*, our specific objective is two-fold. One is to protect both the *locations* and the *identities* of query users during the LBS; The other is to prevent other users’ location privacy from leakage. Therefore, there exists two-level privacy requirements to be considered: personalized user privacy profile and system privacy profile. The former allows a query user to specify his personalized privacy requirement, which is essential for providing anonymous location queries. The latter allows the system to control its privacy protection capability for other users’ locations.

**Personalized user privacy profile** To protect mobile users’ privacy, *location k-anonymity* and *segment l-semantic diversity* are provided to specify their personalized location privacy requirements. Location *k-anonymity* guarantees that it is difficult to identify a specific user among a set of users, based on the CR. Segment *l-semantic diversity* controls that it is difficult to link a user with a specific location semantic (such as a clinic or a church) with high certainty ( $\geq 1/l$ ). For example, in Example 1, if Bob’s CR is  $\{e_2, e_7, e_4\}$ , then an adversary cannot pinpoint the exact semantic of Bob’s walking road. In practice, a mobile user may alter his privacy preference (*k* and *l* values) as often as required. Thus, such privacy requirements should be customizable and provided on a per query basis.

Moreover, our cloaking technique also should not compromise the quality of the LBS (QoS). To guarantee the QoS, e.g., response time, a mobile user *u* should specify some customized requirements. In this framework, *maximum temporal tolerance*  $\sigma_t$  is used to allow a user to specify the critical QoS constraint. It ensures that the temporal delay introduced for waiting to anonymize a query request should be within an acceptable time interval. In summary, the set of parameters (*k*, *l*,  $\sigma_t$ ) consists of *u*’s service profile.



**System privacy profile** In addition, we argue that the protection for other users' location privacy also is essential. Here, one measure is used to specify this type privacy requirement. That is,  $\epsilon$ -differential privacy. It guarantees that its outputs are approximately the same even if a single location record in the dataset is arbitrarily changed. In other words, its outputs are insensitive to the change of any user location. This suggests that the user privacy is protected, and thus, mobile users are not discouraged from participating in the statistical analysis. The level of privacy protection is controlled by the parameter  $\epsilon$ . Lower  $\epsilon$  indicates stronger privacy protection, but also noisier results.

**Threat Model** Generally, in *PrivSem*, we assume that the background knowledge of an adversary is as follows: (i) the cloaking algorithm used by the LA, (ii) all the CRs ever received at the LBS provider, and (iii) the locations of partial mobile users. The first assumption is common in the security community since the data privacy algorithms are usually public. The second assumption states that either the communication channel between the LA and the LBS provider is not secure, or the LBS provider is not trusted (e.g., a commercial organization that collects unauthorized information from its clients for spamming). The third assumption is motivated by the fact that an adversary can pinpoint the locations of some users by the illegal means. If an adversary knows the locations of all users, it is meaningless from the point of location privacy preservation.

### 3.3 Design goals and performance metrics

Protecting user privacy both for query users and other users complicates location anonymization assignment, and may reduce the effectiveness and efficiency of the LBS. Due to the nature of DP, it is possible for a segment containing no users, even though the PSD shows a positive count. Consequently, no users (or an insufficient number thereof) are introduced to protect identity of the query user. Alternatively, the generated CR may be too large, whereas a smaller one is sufficient for location anonymization request. A larger CR deteriorates QoS as well as the efficiency of anonymous queries. Finally, in the non-private location-based queries where the exact location of a query is known, only the required final results are returned. With the privacy awareness, many redundant candidate results may be returned, increasing system overhead. In summary, we focus our attention on the following performance metrics:

**Success Rate** The main purpose of the cloaking algorithm is to maximize the number of query requests perturbed successfully while maintaining their privacy and QoS constraints. Due to PSD data uncertainty, the LA may fail to provide enough protection (e.g., an insufficient users are included). *Anonymization success rate (ASR)* measures the ratio of requests perturbed successfully to the total number of received anonymization requests. The major challenge lies in how to keep ASR close to 100%.

**Relative Levels** In *PrivSem*, the LA no longer utilizes the accurate user count of a segment to compute the CR, the generated CR could tend to larger, resulting in poor QoS and expensive processing cost of anonymous queries. The challenge is to generate small CRs for successful perturbed messages even the accurate user counts of segments are not known. Hence, *relative anonymity level (RAL)* and *relative semantic level (RSL)* are used to measure the performance of generated CRs. More specifically, *RAL* is equal to  $\frac{k_c}{k}$ , and *RSL* is  $\frac{l_c}{l}$ .  $k_c$  and  $l_c$  denote the actual values obtained for the cloaking algorithm.



**System Overhead** Dealing with inaccurate locations increases the complexity of the LBS, which poses scalability issues. An important metric for measuring overhead is the number of candidate results. This quantity affects both the *communication overhead* required to deliver the candidate results from the LBS provider to the LA, as well as the *computational overhead* of the query processing algorithm.

## 4 Spatial cloaking algorithm

To fulfill user privacy requirements and achieve high QoS, the proposed anonymization algorithm is composed of two stages: (i) *segment allocation* and (ii) *online cloaking*. In the first phase, the segments of a road network are roughly grouped into different buckets, so that location anonymization can be performed in a single bucket rather than the entire road network. Using this information, the second phase locates the bucket of the segment of each query user and anonymizes the segment based on user privacy profile. In this section, we present the segment allocation, then detail the online cloaking in the next section.

### 4.1 Segment allocation

This phase mainly allocates the segments of a road network to different buckets according to users' privacy requirements. To capture most user privacy requirements, we make the following observation.

**Observation 1** The location semantic privacy requirements  $L$  of user privacy profiles follow a Gaussian distribution  $L \sim N(\mu, \sigma^2)$ , i.e., most of user location semantic privacy requirements fall in the middle range, and fewer have higher privacy requirements. The parameter  $\mu$  is the mean of the distribution, and the parameter  $\sigma$  is its standard deviation.

It follows that we can make use of the 68-95-99.7 empirical rule, also known as the  $3\sigma$  rule, which states that about 99.7% of values sampled from a Gaussian distribution lie within three standard deviations away from the mean. With this fact, it is often sufficient to set the appropriate semantic number of a bucket at  $\mu + 3\sigma$  for satisfying almost all user location anonymization in a single bucket. Definition 4 formally elaborates the objective of segment allocation.

**Definition 4** (*Segment Allocation*) The segments of a road network  $G = (V, E, \xi)$  are allocated to  $p$  buckets,  $G_1, G_2, \dots, G_p$ , where  $p > 1$  and  $G_i = (V_i, E_i, \xi_i)$ , such that  $V = \bigcup_{1 \leq i \leq p} V_i$ ,  $E = \bigcup_{1 \leq i \leq p} E_i$ ,  $\xi(E) = \bigcup_{1 \leq i \leq p} \xi_i(E_i)$ , and the following conditions are satisfied.

- (i) The segments of all buckets are disjoint, i.e.,  $\forall 1 \leq i, j \leq p, E_i \cap E_j = \emptyset$ .
- (ii) The semantic number of a bucket must exceed the threshold  $\mu + 3\sigma$ , i.e.,  $|\xi(E_i)| = |\bigcup_{e \in E_i} \xi(e)| \geq \mu + 3\sigma$ .

The cloaking algorithm aims at protecting the location privacy of mobile users. Besides, it should not compromise QoS, which depends mostly upon maximum temporal tolerance and system overhead. As mentioned, the number of candidate results is used to measure system overhead, which is formulated in Definition 5. Without loss of generality, we focus our discussion on  $K$ -nearest neighbors ( $KNN$ ) queries.

**Definition 5** (*LBS Server Processing*) [7, 45] For a query  $q$  with associated a CR  $S_c$ , the candidate results of  $q$  consists of two parts: (1) the POIs on the segments of  $S_c$ , and (2) the results as  $q$  are issued on the boundary nodes of the boundary set  $S_{bn}$ , where the boundary set is a set of nodes whose some connected edges are not included in  $S_c$ . Formally,  $Cost(q, S_c) = (\bigcup_{s \in S_c} O(q, s)) \cup (\bigcup_{v \in S_{bn}} O(q, v))$

With this query processing model, it can be observed that the system overhead  $Cost(q, S_c)$  is significantly affected by parameters  $|S_c|$  and  $|S_{bn}|$ . However, decreasing  $|S_c|$  and  $|S_{bn}|$  imposes conflicting demands on  $Cost(q, S_c)$ . The reason mainly involves the fact that segments that are near each other tend to possess similar semantic labels. For a user privacy profile, our goal is to find the optimal CR which is minimized in terms of system overhead, while satisfying location  $k$ -anonymity and segment  $l$ -semantic diversity. To sum up, our problem is equivalent to the following optimization problem:

$$\text{Minimize } Cost(q, S_c), \text{ subject to } Count(S_c) \geq k, |\xi(S_c)| = |\bigcup_{e \in S_c} \xi(e)| \geq l.$$

As shown in the paper [45], the problem of computing an optimal CR is NP-hard.

**Solution** Given the analysis above, we develop a greedy solution called *EIRank*. An intuitive guideline is that adjacent segments with different semantic labels should be cloaked together in order to provide a compact structure and semantic preference simultaneously. Under this guideline, we prefer cloaking the segments exhibiting *structure similarity* and *semantic label dissimilarity*. To measure the similarity of linkage structures and the dissimilarity of semantic labels, we present two scoring functions:  $S(n_1, n_2)$  and  $Diff(e_p.\varphi, e_q.\varphi)$ .

In many applications, objects are deemed similar if they are related to similar objects. Motivated by this intuition, a general similarity metric called SimRank is naturally adapted to capture the similarity of linkage structures. The calculation of SimRank is given in (2).

$$S(n_1, n_2) = \begin{cases} 1 & n_1 = n_2 \\ \frac{C}{|I_{n_1}| |I_{n_2}|} \sum_{j \in I_{n_2}} \sum_{i \in I_{n_1}} S(i, j) & n_1 \neq n_2 \end{cases} \tag{2}$$

In this equation, the parameter  $C$  refers to as a decay factor, is a constant between 0 and 1, and the parameter  $I_n$  denotes the neighboring set of  $n$ . Note that (2) is set to 0 when  $I_{n_1} = \emptyset$  or  $I_{n_2} = \emptyset$ .

To evaluate the dissimilarity of semantic labels of segments, we utilize the normalized edit distance. In this case, the dissimilarity of semantic labels  $Diff(e_p.\varphi, e_q.\varphi)$  is measured by the edit distance between the semantic labels in regard to the length of the semantic label. The edit distance,  $Edit(e_p.\varphi, e_q.\varphi)$ , between two semantic labels,  $e_p.\varphi$  and  $e_q.\varphi$ , is defined as the minimum number of basic operations required to transform one semantic label into the other. In this paper, the basic operations are referred as insertion, deletion and substitution of symbols, which is formalized as follows.

Let  $T_i(a)$  denotes the insertion of symbol  $a$ ,  $T_d(a)$  denotes the deletion of symbol  $a$ , and  $T_s(b|a)$  denotes the substitution of symbol  $a$  by symbol  $b$  ( $a \neq b$ ). Then,

$$Diff(e_p.\varphi, e_q.\varphi) = \frac{Edit(e_p.\varphi, e_q.\varphi)}{Max(|e_p.\varphi|, |e_q.\varphi|)} \tag{3}$$

where  $e_p.\varphi$  represents the label function of  $e_p$ , and  $Max(|e_p.\varphi|, |e_q.\varphi|)$  is a function that computes the larger length of the two labels  $e_p.\varphi$  and  $e_q.\varphi$ .

To integrate semantic information and linkage structure collaboratively for segment allocation, we develop a greedy solution, referred as *EIRank*, for simultaneously representing link-based similarity and semantic-based dissimilarity. At a high level, our solution consists

of four steps: *EI network construction, label clustering, augmented EI network construction and segment allocation*. Below, we will introduce each of them in details.

**EI Network Construction** For ease of exposition, the semantic label of each edge is unique in this paper. To combine linkage structure and segment semantic, an edge interaction (EI) network is firstly transformed from a semantic road network. More specifically, an EI network node (*e-node*), represents an edge in the original semantic road network, and two e-nodes are said to be *adjacent* if their corresponding edges share a common node in the original semantic road network. Additionally, the labels of e-nodes in the EI network are provided by the semantic labels of the corresponding edges in the road network. For instance, the edges  $e_1$  and  $e_2$  have a common node  $v_2$  in the semantic road network (Figure 3a), and thus the e-nodes  $e_1$  and  $e_2$  are linked together in the EI network (Figure 3b). Furthermore, the segment id itself is sufficient to indicate the semantic label of a segment, so that the labels of the e-nodes in the EI network are omitted to mark.

**Label Clustering** The problem of calculating the dissimilarity of two segment labels is translated into the equivalent one of calculating the dissimilarity of two e-node labels in the EI network. As mentioned before, we are able to use the normalized edit distance to accomplish the goal. In the case of the labels of the two e-nodes  $e_3$  and  $e_4$  in Figure 3b, by performing the basic operations  $T_s(l|h)$ ,  $T_s(b|r)$ ,  $T_d(c)$  and  $T_d(h)$ , the label of e-node  $e_3$  is transformed to the label of e-node  $e_4$ . Thus, the dissimilarity of these two e-node labels is  $Diff(e_3.\varphi, e_4.\varphi) = \frac{2}{3}$ .

Based on the dissimilarity of the labels of e-nodes, we then perform a generalized  $k$ -medians clustering [32] for these e-node labels. In Figure 3b, the result of label clustering is  $\{church, police, park\}$  and  $\{bar, club\}$ .

**Augmented EI Network Construction** In the third step, we create a virtual node for each label cluster and connect the e-nodes whose labels are in the same cluster to the virtual node. The new generated network is called *augmented EI network*. Through adding the virtual nodes, the e-nodes in the same label cluster tend to have higher structure similarities. For example, Figure 3c gives the augmented EI network corresponding to Figure 3b. Two virtual e-nodes  $o_1$  and  $o_2$  are created to represent the clusters  $\{church, police, park\}$  and  $\{bar, club\}$ , respectively. In particular, the virtual node  $o_1$  is connected to the e-nodes in the set  $\{e_1, e_2, e_3, e_8, e_9, e_{10}\}$ . In the same manner, the e-nodes in the set  $\{e_4, e_5, e_6, e_7\}$  are connected to the virtual node  $o_2$ .

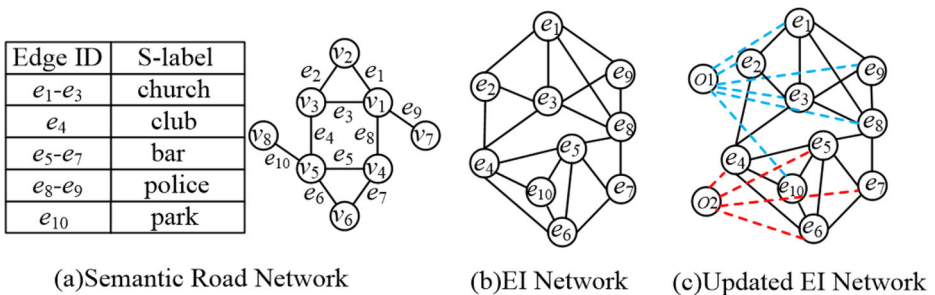


Figure 3 Example of EIRank strategy

**Segment Allocation** As stated earlier, the segments of a CR needs to be structurally similar and semantically dissimilar. Based on the first three steps, the dissimilarity of the e-node labels is converted to the similarity of the linkage structure. It is consistent with the similarity of the linkage structure. Next, the function  $S(e_p, e_q)$  is employed to evaluate the similarity for every pair of non-virtual e-nodes.

To calculate SimRank more efficiently, we adopt the method in [13]. In such a case, the similarity of e-nodes is measured by (4), which states that the similarity of two e-nodes is the expectation of the total time which is the time taken by two random walkers starting from two different nodes to finally meet.

$$S(e_p, e_q) = E(C^{\tau(e_p, e_q)}) \tag{4}$$

Once the similarity of all e-node pairs has been calculated, we leverage the single-linkage hierarchical clustering [39] to perform the segment allocation. The function  $\text{Allocate}(e_p, e_q, G_S)$  is used to describe this process.

The complete description of our EIRank strategy is shown in Algorithm 1.

---

**Algorithm 1** EIRank algorithm.

---

**Input:** Semantic road network  $G = (V, E, \xi)$ , Bucket scale  $N_l$

**Output:** Buckets  $G_1, G_2, \dots, G_p$

- 1 Transform the  $G$  into EI network;
  - 2 Execute the label clustering for e-nodes;
  - 3 Compute  $S(e_p, e_q)$  for all e-node pairs;
  - 4  $\text{Allocate}(e_p, e_q, G_S)$ ;
  - 5 Merge buckets  $G_i$  where  $|\xi(G_i)| < N_l$ ;
  - 6 Return non-empty buckets  $G_1, G_2, \dots, G_p$ ;
- 

## 4.2 Ordered locating index

Due to the daunting size of segments, it is costly and time-consuming to search the position of a segment in a segment allocation. To fast and efficiently for performing this search, we devise a novel data structure-Ordered Locating Index.

**Ordered Locating Index (OLI)** For a particular segment, this data structure allows for high efficient and fast computation of the position in the segment allocation. It organizes the segments in order. Each record is represented as  $(Sid, Bid, Pid, Pointer)$  where Sid is the segment identifier, Bid is the bucket identifier of the segment Sid, Pid is the position identifier of segment Sid in bucket Bid, and Pointer is a pointer to the next record. With the mapping relation, we can quickly locate the position of a segment in a segment allocation. More precisely, (5) is used to compute the sequence of segment  $Seq(e_{i,j})$  in the ordered linked list to obtain the Bid and Pid of  $e_{i,j}$ . In the equation, the symbol  $e_{i,j}$  indicates a segment that connects the nodes  $i$  and  $j$ . Meanwhile, the first three entries are mainly used to compute the total number of segments before the segment  $e_{i,j}$ .

$$Seq(e_{i,j}) = \sum_{k=1}^{i-1} degree(k) - |S_{overlap}|_{S_{overlap}=\{e_{l1}, t < i, l < i} + |S_{prior}|_{S_{prior}=\{e_{ip}, i < p < j} + 1 \tag{5}$$

Take the segment  $e_{4,6}$ (i.e.,  $e_7$ ) in Figure 1 as an example. According to the (5), its segment sequence is  $Seq(e_{4,6}) = degree(v_1) + degree(v_2) + degree(v_3) - (|\{e_{1,2}, e_{2,3}\}|) + |\{e_{4,5}\}| + 1$

= 2+3+2-2+1+1 = 7. Then, search for the 7th record in OLI which is shown in Figure 6, we derive that Bid = 2 and Pid = 8. It means that the segment  $e_{4,6}$  is in bucket 2, at position 8.

### 5 Online cloaking

In the previous section, we have elaborated the first phase of the proposed approach. Once partitioned buckets have been obtained, the remaining work is to generate a CR according to a user’s online request. As illustrated in Example 1, to protect other users’ privacy, we cannot use the real count of mobile users in each segment. Instead, the PSD is leveraged to publish this count.

#### 5.1 Building the user PSD

This step consists of building a user PSD (at the CSP component) to be later used for location anonymization assignment at the LA. Building the PSD is a vital step, because it determines how accurate the released data is, which in turn affects ASR. In this part, a *naive user PSD* for mobile users on a road network is firstly presented. Then, inspired by the data sparsity, we then modify it to release more accurate counts.

**Naive User PSD** To construct user PSD, we devise the *Segment User Count Map*(SUCM) as the underly fundamental structure to record a count of the number of mobile users located in each segment. Each entry is defined as a tuple ( $Sid, N$ ) where  $Sid$  is the segment identifier while  $N$  is the number of mobile users on this segment. This structure is dynamically maintained to keep track of the total number of mobile users within each segment. In addition to the mapping of each segment to its current count, we also devise a hash table  $HT$  to keep track of current locations of mobile users. Each entry in  $HT$  for a mobile user is represented as ( $Uid, Sid$ ), where  $Uid$  is the mobile user identifier, and  $Sid$  is the segment identifier where  $Uid$  is located. This structure allows for efficient and fast computation of mobile user counts belonging to a particular segment of a road network. Figure 4 illustrates these two data structures.

With this fundamental spatial structure, we need to build a user PSD (donated by  $PSD_u$ ) according to DP. It only requires one simple step to fulfill the construction of  $PSD_u$ . That is, the noisy counts of segments are computed by directly injecting random Laplace noise

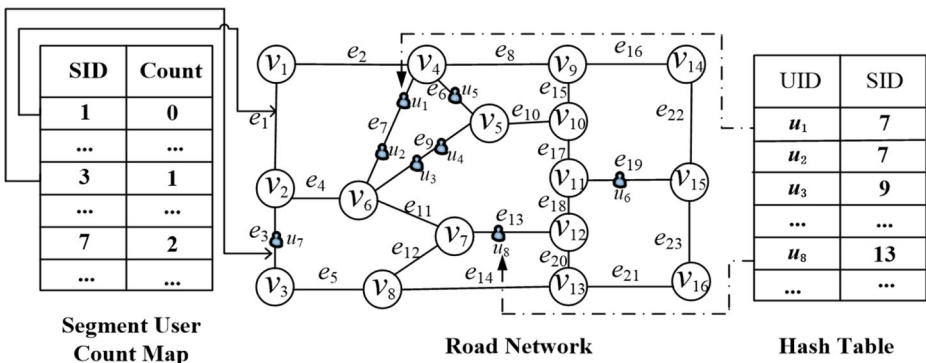


Figure 4 An illustrate of proposed fundamental data structure for PSD

with scale  $\lambda = \frac{2}{\epsilon}$  to the actual counts of these segments. Typically, all segments have non-overlapping extents, which yields a constant and low sensitivity of 2 (i.e., changing the location of any one specified user in the dataset may affect at most the counts of two segments). Thus, injecting random Laplace noise  $Lap(\frac{2}{\epsilon})$  satisfies  $\epsilon$ -differential privacy.

**Customized User PSD** In real life, compared to the total number of segments in a road network, the total number of mobile users is very small. The majority of segments have very low to zero count. This data sparsity issue is an immense challenge for privacy-preserving techniques since the injected noise is more likely to dominate the released counts in presence of a small set of mobile users. In other words, when each segment is perturbed individually, it will result in high relative error for the released counts of sparse segments due to the injected perturbation noise. Inspired by this, we propose *adaptive group mechanism* to mitigate the data sparsity issue.

The key idea is that segments with small statistics should be grouped together if they have close statistics and similar statistical trends. Due to the spatial correlation, it is very likely that adjacent segments belong to the same area (such as suburb, city center, etc). Therefore, they possess similar constraints on the user counts, leading to more similar statistical properties. For example, a *collector* road restricts the number of users to a low value whereas an *expressway* attracts a considerably higher number of users; Besides, adjacent roads of a congested road also tend to be congested. To utilize these heuristics, we propose to use the *structure information* to characterize the statistical trend. Let  $S(e_i)$  denote the neighboring set of a segment  $e_i$ . We then adopt (6) to measure the similarity of statistical trends. The expression  $signal(e_i, e_j)$  is used to check whether they are roads of the same type. If they both are *expressway* (*collector* etc), the value is 1; otherwise, it is 0.  $J(e_i, e_j)$  is the Jaccard similarity coefficient between  $S(e_i)$  and  $S(e_j)$ . Eventually, segments with small statistics and high similarity are grouped together.

$$SJ(e_i, e_j) = \alpha \times signal(e_i, e_j) + (1 - \alpha)J(e_i, e_j) \tag{6}$$

Figure 5 describes our customized user PSD mechanism. As observed, it is further decomposed into  $M_1$ ,  $M_2$  and  $M_3$ , which operate sequentially.  $M_1$  performs a sparse computation algorithm between noisy count  $\tilde{nc}_{e_i}$  of the segment  $e_i$  and the noise resistance threshold  $\tau_1$ . The result of the calculation is forwarded to  $M_2$ , which makes use of it to decide whether to group a segment separately or group with other segments. It means  $M_2$  determines the final groups (called *partitions*) of segments. Once the partition structure of the segments is established,  $M_3$  injects the Laplace noise to each partition to ensure differential privacy. Since we assume the uniform data distribution within each partition, the noisy count of each segment can be estimated with the average partition count.

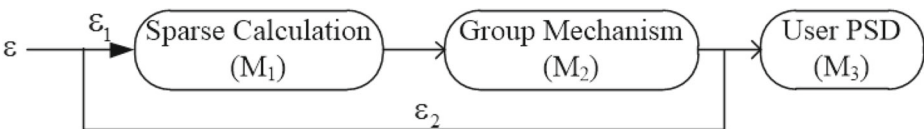


Figure 5 Internal mechanics of customized user PSD

The sub mechanisms  $M_1$  and  $M_3$  can be performed directly, as they only depend on Laplace mechanism. Next, we elaborately describe the procedures of group strategy  $\mathfrak{R}$ . Initially, let the segment  $e_j$  itself as an independent partition if  $\widetilde{nc}_{e_j} > \tau_1$  and add the partition to  $\mathfrak{R}$ ; Then, sort all segments not in  $\mathfrak{R}$  in order of increasing  $\widetilde{nc}_{e_j}$ , denoted by  $\Omega$ . Subsequently, as long as  $\Omega$  is not empty, we perform the following operations: initialize a new partition(e.g.,  $g$ ) with the first segment  $e_{(1)}$ , and check the next segment  $e_{(k)}$  in  $\Omega$ . If the distinction between the noisy counts of  $e_{(1)}$  and  $e_{(k)}$  is less than  $\tau_2$ , and the sum of noisy counts of the segments in  $g$  is less than  $\tau_1$ , we calculate their similarity. If this value is larger than  $\tau_3$ , remove  $e_{(k)}$  from  $\Omega$  and place it into the partition  $g$ , otherwise, skip this segment and do the same check for the next segment. For other cases, add  $g$  to  $\mathfrak{R}$  and remove  $e_{(1)}$  from  $\Omega$ . Finally, the final group strategy  $\mathfrak{R}$  is obtained.

Note that, the threshold  $\tau_2$  is the error threshold that decides whether the statistics of two segments are close to each other, and  $\tau_3$  is the similarity threshold that decides whether the statistic properties of two segments are similar. Both  $M_1$  and  $M_3$  must be private, as they have access to sensitive dataset  $D$ . Moreover, any post-processing of differentially private data remains differentially private, therefore  $M_2$  does not violate privacy. Let  $\epsilon_1$  and  $\epsilon_2$  be the budgets spent in  $M_1$  and  $M_3$ , respectively. Then, due to sequential theorem(Theorem 2), the privacy budget of adaptive group mechanism is  $\epsilon_1 + \epsilon_2 = \epsilon$ .

*Example 3 (Adaptive Group Mechanism)* Suppose there are three segments  $\{e_1, e_2, e_3\}$  needed to be grouped, and their similarities are  $SJ(e_1, e_2) = 0.92$ ,  $SJ(e_1, e_3)=0.35$  and  $SJ(e_2, e_3)=0.45$ . Let  $\tau_1=50$ ,  $\tau_2=20$  and  $\tau_3=0.8$ . The noisy counts for these three segments are 15.3, 9.7 and 65.2, respectively. Since  $65.2 > 50$ , the segment  $e_3$  is a separate group and is added to  $\mathfrak{R}$ . For the segments  $e_1$  and  $e_2$ , their statistical similarity is 0.92. As  $0.92 > 0.8$ , and  $15.3 - 9.7=5.6$  is smaller than 20, we can group these two segments together. Thus, the final group strategy is  $\mathfrak{R} = \{\{e_3\}, \{e_1, e_2\}\}$ .

**Error Analysis** To protect the user privacy, the exact location of a query user is usually extended to a larger CR which needs to be meet the user privacy requirement. Due to the nature of DP, a CR may not contain enough mobile users even if the released count exceeding the specified  $k$ . To ensure high ASR, we should guarantee the real count of a CR is larger than  $k$  with high possibility. To achieve this goal, we analyze the error in the count reported by a CR. Indeed, the noisy count of a CR is computed as the sum of noisy counts of the partitions which are contained in the CR. Below we quantify the noise accumulated in the process, which will help us to improve ASR of the cloaking algorithm.

Formally, let  $\widetilde{nc}_{cr}$  be the count released by the PSD for a CR, and donate by  $nc_{cr}$  its real count. Let  $n_1$  and  $n_2$  denote the number of *complete partitions* and *partial partitions* contained in the CR, respectively. Given a partial partition  $g_p$  and its corresponding complete partition  $g_c$ , the expected error is  $Error(g_p) = \sum_{e_i \in g_p} (\widetilde{nc}_{e_i} - nc_{e_i}) = \sum_{e_i \in g_p} (\frac{\sum_{e_j \in g_c} (nc_{e_j})}{|g_c|} + \frac{1}{|g_c|} Lap(\frac{2}{\epsilon}) - nc_{e_i}) = \underbrace{(\frac{\sum_{e_j \in g_c} (nc_{e_j})}{|g_c|} |g_p| - nc_{g_p})}_{\text{Approximation error}} + \underbrace{\frac{|g_p|}{|g_c|} Lap(\frac{2}{\epsilon})}_{\text{Laplace error}}$ .

As observed, the error of a partial partition is composed of approximation error and laplace noise error. Since the approximation error is data-dependent and difficult to quantify, we roughly use the introduced noise error to measure the error of a partial partition. In



contrast, the error of a completed partition is totally composed of noise error. Then, we can write  $\widetilde{nc}_{cr}$  as follows.

$$\widetilde{nc}_{cr} = nc_{cr} + \sum_{i=1}^{n1} Lap(\frac{2}{\epsilon}) + \sum_{j=1}^{n2} ratio_j \times Lap(\frac{2}{\epsilon}) = nc_{cr} + \sum_{i=1}^n Lap(\frac{2}{\epsilon}) \quad (7)$$

In this equation,  $n$  is the sum of  $n_1$  and  $\lceil \sum_{j=1}^{n2} ratio_j \rceil$ . For each partial partition  $g_p$ ,  $ratio$  is the ratio of the segment number contained in  $g_p$  to that of its associated completed partition  $g_c$ , i.e.,  $\frac{|g_p|}{|g_c|}$ . Let the random variable  $Y = \sum_{i=1}^n Lap(\frac{2}{\epsilon})$  denotes the sum of these Laplace noises. It thus determines the error in estimating the absolute value of the count  $\xi = \| \widetilde{nc}_{cr} - nc_{cr} \|_1$ . The following corollary gives a formal description about this quantity.

**Corollary 2** (*Error Bound for CR Counting*) *For any count query for a CR, the noisy count  $\widetilde{nc}_{cr}$  obtained by summing the noisy counts of the  $n$  partitions contained in the CR, with probability at least  $1 - \delta$ , the quantity  $\xi = \| \widetilde{nc}_{cr} - nc_{cr} \|_1$  is at most  $O(\frac{2}{\epsilon} \sqrt{n \log \frac{1}{\delta}})$ .*

*Proof(sketch).* The proof follows from Corollary 1, where we choose  $v = \sqrt{\sum_{i=0}^{n-1} (\frac{2}{\epsilon})^2} \sqrt{2 \ln \frac{2}{\delta}}$ .

### 5.2 Cloaking algorithm

In this subsection, we present our online cloaking algorithm, which is sketched in Algorithm 2. It involves six main inputs:  $u$  (mobile user),  $(x, y) \in e_i$  (user location),  $(k, l, \sigma_i)$  (user profile),  $H$  (hash table), OLI (ordered locating index), and  $PSD_u$  (user PSD).

The algorithm starts by performing the initialization, after which it extracts some basic information (Lines 1-2). Subsequently, it leverages a semantic based cloaking function to discover the semantic-based cloaked area  $CR_u$  (Line 3). At this step, the algorithm ignores the constraint of  $k$ , but rather focuses on the semantic requirement. Then, it calculates the count of mobile users in  $CR_u$  (Lines 4-5). At this step, to protect other users' location privacy, the noisy count of each segment is used, rather than the real count. Next, the algorithm analyzes the lower bound for the real user count of  $CR_u$ , that is,  $\widetilde{nc}_{CR_u} - \xi$ . Finally, it checks whether this value satisfies location  $k$ -anonymity constraint. If the current  $CR_u$  is  $k$ -anonymity, it stops. Otherwise, it recursively selects the neighboring segment with largest noisy count to add into  $CR_u$  until  $CR_u$  satisfies  $k$ -anonymity (Lines 6-9). Note that, If the current largest noisy count is less than  $\frac{2\sqrt{2}}{\epsilon}$ , the algorithm stops the cloaking process and returns failure.

During the process, some subtle for processing noisy counts are introduced. Specifically, if the noisy count of a PSD node containing current segment is less than  $\frac{2\sqrt{2}}{\epsilon}$ , its value is set to zero. Recall that the major purpose of the cloaking algorithm is to achieve high ASR. In that sense, we desire to ensure that the user count of a selected segment is non-empty, i.e., the real user count of a segment is strictly positive. Given the Laplace mechanism of DP, each PSD node count is the sum of noisy and real count. For the distribution of injected noise, it has standard deviation  $\mu = \frac{2\sqrt{2}}{\epsilon}$ . Hence, if the count of a PSD node is less than  $\mu$ , then with high probability it is empty. Based on this analysis, we prefer to set the noisy counts of segments in a such PSD node to 0, further increasing ASR.

**Algorithm 2** Online cloaking algorithm.

---

**Input:** user  $u$ , location  $(x, y) \in e_i$ , user profile  $(k, l, \sigma_t)$ , hash table  $H$ , ordered locating index  $OLI$ , user PSD  $PSD_u$

**Output:** the cloaking area  $CR_u$

- 1 initialize  $CR_u \leftarrow \Phi$   $\widetilde{nc}_{CR_u} = 0$ ;
- 2  $Sid_u = H(u)$ ;  $(Bid_u, Pid_u) = OLI(Sid_u)$ ;
- 3  $CR_u = \text{Semantic\_Cloaking}(l, Bid_u, Pid_u)$ ;
- 4 **for**  $i \leftarrow 1$  **to**  $|CR_u|$  **do**
- 5    $\widetilde{nc}_{CR_u} = \widetilde{nc}_{CR_u} + \widetilde{nc}_{Sid_i}$ ;
- 6 **while**  $\widetilde{nc}_{CR_u} - \xi < k$  **do**
- 7    $Sid_j \leftarrow$  selected neighbor segment with largest noisy count;
- 8   insert  $Sid_j$  into  $CR_u$ ;
- 9    $\widetilde{nc}_{CR_u} = \widetilde{nc}_{CR_u} + \widetilde{nc}_{Sid_j}$ ;
- 10 Return  $CR_u$ ;

**Function:**  $\text{Semantic\_Cloaking}(l, Bid_u, Pid_u)$

- 1 **if**  $(l = 1)$  **then**
- 2    $CR_u = Sid_u$ ;
- 3 **else**
- 4    $S_{sem} \leftarrow \Phi$ ;  $Pos_c = 0$ ;
- 5   **while**  $Pos_c < Pid_u$  **do**
- 6      $Pos_o = Pos_c$ ;
- 7     update  $Pos_c = \text{Cloak}(l, Pos_c + 1, Bid)$ ;
- 8     **if**  $\text{residual\_semantic}(Bid, Pos_c) < l$  **then**
- 9        $Pos_c = CL(Bid_u)$ ;
- 10   insert  $Cid(\text{loc}(Pos_o, Pos_c])$  into  $CR_u$ ;

---

For the semantic based cloaking function, its core lies in making use of the *segment oblivious* property [29]. When  $l = 1$ , it returns the segment of user location as the CR. It suggests that user current segment is sufficient to satisfy the semantic privacy requirement. For other situations, it consists of a number of iterations. In each iteration, it detects a cloaked segment set (short for cloaked set)  $S_L$  from the first unprocessed location  $loc_{pos_c+1}$  in bucket  $Bid_u$ . The formed cloaked set satisfies  $l$ -semantic diversity. Then, we check the number of remaining semantics in this bucket. When it is less than  $l$ , these remaining segments are also put into  $S_L$ . Next, determine whether  $Sid_u$  is in  $S_L$ . If not, this function continues to iteratively detect  $S_L$  in the same manner until it find the required  $S_L$ . Finally, for each segment  $e \in S_L$ , we insert it into  $CR_u$ . During the process, the notation  $Pos_o$  represents the last position of  $S_L$  which is detected in the previous iteration, and  $Pos_c$  represents the last position of  $S_L$  which is detected in the current iteration.

*Example 4 (Semantic Based Cloaking)* Suppose the content of a bucket for Figure 1 is  $\{e_3, e_1, e_4, e_{11}, e_2, e_6, e_9, e_7, e_{22}, e_{17}, e_{12}\}$ , and the semantics of these segments are shown in Figure 6. Two users  $u_1$  and  $u_2$  have the same semantic privacy requirement  $l=3$ , and they locate in the segment  $e_{22}$  and  $e_1$ , respectively. Initially, the algorithm traverses the bucket from the scratch (i.e.,  $Pos_o=0$ ) and finds the first 3-semantic diversity cloaked set  $S_L = \{e_3, e_1, e_4\}$ . Next, it checks the number of remaining semantics in this bucket (i.e., 6). Because 6 is larger than 3, and then  $S_L = \{e_3, e_1, e_4\}$  is a cloaked set (i.e.,  $Pos_c=3$ ). Since

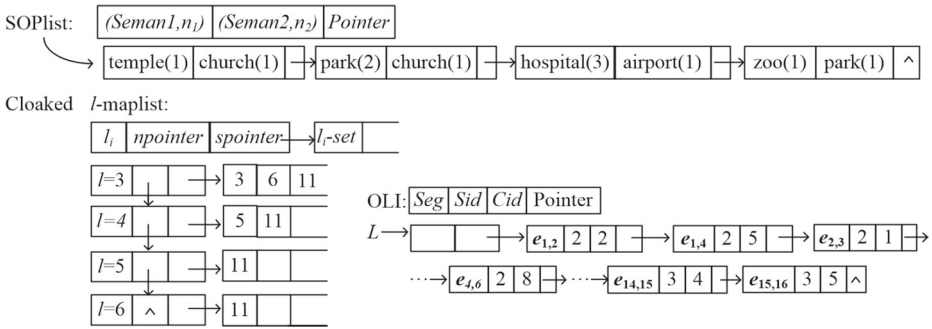


Figure 6 An illustrate of SOplist and Cloaked  $l$ -maplist

the segment  $e_{22}$  is not in  $\{e_3, e_1, e_4\}$ , The algorithm continues to search the next cloaked set  $S_L = \{e_{11}, e_2, e_6\}$  in the same manner ( $Pos_o=3, Pos_c=6$ ). When it retrieves the cloaked set  $S_L = \{e_9, e_7, e_{22}, e_{17}\}$ , and detects that the remaining semantics number is 1, which is smaller than 3. Thus, the segment  $e_{12}$  needs to be added into  $S_L$ . At this step, the segment  $e_{22}$  is in  $S_L = \{e_9, e_7, e_{22}, e_{17}, e_{12}\}$ . Therefore, the cloaked set  $S_L = \{e_9, e_7, e_{22}, e_{17}, e_{12}\}$  is  $u_1$ 's semantic based CR. To compute the semantic based CR for  $u_2$ , the algorithm just needs to perform the first iteration as the user  $u_1$ , and get  $S_L = \{e_3, e_1, e_4\}$ .

### 5.3 Optimizations

Despite its simplicity, the basic version of *PrivSem* introduced above suffers several drawbacks. (1) For anonymizing each query request, the function *Semantic\_Cloaking()* recalculates the basic semantic-based CR from the scratch every time, which deteriorates the efficiency of the LA. (2) It attempts to execute anonymization immediately after a new query request arrives, i.e., the LA processes each query request completely independently. It is expected that a lots of attempts are required to process these requests, thereby incurring the scalability problem. In what follows, corresponding to these drawbacks, we develop a series of optimization strategies to improve the performance of the LA.

**Recording semantic cloaking (RSC)** To facilitate the execution of the semantic-based cloaking algorithms, we also devise *SOplist* and *Cloaked l-maplist* these two other data structures. The former is a 2-semantic diversity index, and it aims to speed up the computation of the basic semantic-based CR. Each record of SOplist is in the form  $((\text{seman1}, n_1), (\text{seman2}, n_2), \text{Pointer})$ , where  $(\text{seman1}, n_1)$   $(\text{seman2}, n_2)$  indicates  $n_1$   $(n_2)$  adjacent segments of semantic label  $\text{seman1}$   $(\text{seman2})$ , while Pointer is a pointer to the next record.

Cloaked  $l$ -maplist is designed to record the CRs that have been generated for distinct semantic requirements so far. It is achieved by re-using the mapping between segments and CRs. A basic cell of Cloaked  $l$ -maplist is represented as  $(l_i, npointer, spointer)$  and  $l_i\_set$ , where  $l_i$  denotes  $l_i$ -semantic diversity,  $npointer$  and  $spointer$  are pointers to the next basic cell and  $l_i\_set$ , respectively, and  $l_i\_set$  records the last position of each CR regarding semantic requirement  $l_i$ .  $l_i\_set$  is dynamically maintained to keep track of the current maximum position of CRs of semantic requirement  $l_i$  in a bucket.

*Example 5 (SOplist and Cloaked l-maplist)* Continuing with Example 4, Figure 6 shows the SOplist and Cloaked  $l$ -maplist corresponding to the bucket. In Example 4, to compute the

semantic based CR for  $u_2$ , the algorithm recalculates from the scratch, which deteriorates the efficiency of the LA. With the help of Cloaked  $l$ -maplist, instead of recalculating it, the algorithm examines whether this cloaked set has been formed before. After deriving  $u_1$ 's semantic based CR  $S_L = \{e_9, e_7, e_{22}, e_{17}, e_{12}\}$ , the maximum of  $l_3$ -set is updated as 11. From the structure OLI, it is easy to conclude that the position  $Pid_{u_2}=2$ . Since 2 is smaller than 11, this means that its semantic based CR has been formed before, and thus the algorithm can directly obtain it without recalculating. As  $2 < 3$ , the last position of first cloaked set, then the semantic based CR is generated by adding the segments in interval  $(0,3]$ , that is  $S_L = \{e_3, e_1, e_4\}$ .

**Delay and sharing anonymizing (DSA)** To increase the scalability of LA, we propose *delay and sharing anonymizing* strategy. It explores the possibility of sharing processing in the location anonymization operation, by combining query requests with nearby locations together and perturbing them as an entirety. This makes sense, because for each query request, one can wait for a period of time  $T_w$ , before starting the anonymization process. And besides, the parameter  $T_w$  is set according to users' maximum tolerable  $\sigma_t$  in user profile. Through the sharing anonymization, it can significantly reduce the burden on LA, further increasing its scalability.

Concretely, given a set of query requests  $\{q\}_{i=1}^t$  with corresponding users' privacy profiles as  $\{k_i, l_i, \sigma_i\}_{i=1}^t$  and arriving time  $\{r_i\}_{i=1}^t$ . These query requests are pushed to the query queue  $Q_q$  in a increasing order of  $(r_i + \sigma_i)$ . Once the first query request  $q_i$  is popped up from  $Q_q$ , the algorithm attempts to perform anonymization. First, it calculates and initiates the cloaked area  $CR_u$  based on `Semantic_Cloaking()`. Then, it traverses  $Q_q$  and find the request set  $S_{qr}$  whose semantic requirements are similar to  $q_i$ , i.e., for  $\forall q_j \in S_{qr}, l_j = l_i$ . Next, for each query request  $q_j \in S_{qr}$ , it checks whether its issuing segment  $e_j$  is in  $CR_u$ . If  $e_j \in CR_u$ ,  $q_j$  is deleted from  $Q_q$ . Besides, the parameter  $k_{max}$  is used to record the maximum  $k$  associated with requests in this anonymization. Finally, based on the noisy count of  $CR_u$  and  $k_{max}$ , it extends  $CR_u$  until it contains enough mobile users.

## 6 Privacy analysis

In this section, we give the formal analysis about privacy guarantees of *PrivSem* for both query users and other users. From the perspective of query users, it needs to consider the resilience of the location anonymization against an adversary's attack: based on his prior knowledge and understanding concerning the anonymization model, the adversary attempts to pinpoint query users' identity, location and location semantic through the perturbed information. For the other users, it should guarantee that it is difficult to link a specific segment or region with these users. It is noteworthy that the attack discussed here focuses on one-shot queries.

**Privacy guarantee for query users** Given a CR of a query user  $u$  as a set of segments  $CR_u$ , the expected identity protection is provided if at least  $k$  users are indistinguishable to the adversary. For location protection, compared to  $l$ -segment diversity, we argue that segment  $l$ -semantic diversity provide more stronger privacy protection capability. That is, at least  $l$  segments in  $CR_u$  are indistinguishable to the adversary. Alternatively, it is difficult to pinpoint the exact semantic of query location. Thus, from the adversary's perspective, each user  $u$  is associated with this query request with equal probability which is no larger than  $\frac{1}{k}$ ; the probability of  $u$  associated with each segment is no more than  $\frac{1}{l}$ , and the semantic

number associated with  $CR_u$  is no less than  $l$ . However, with effective attacks, the adversary can identify that these associations have much higher probability than required, thereby disclosing  $u$ 's privacy with high confidence. Thus, the notion of *Linkability* is proposed to capture such vulnerability.

**Definition 6** (Linkability) Given a user  $u$  issues a query  $q$  with exact location as  $v_{loc}^* \in e_u$ , location semantic as  $e_u.\varphi$  and anonymous CR as a set of segments  $CR_u$ . Based on  $CR_u$  and background knowledge  $K_b$ , the identity linkability  $Pr[q \leftarrow u] | CR_u, K_b$  is the probability an adversary can infer  $u$ 's association with  $q$ , and location linkability  $Pr[u \leftarrow e_u | CR_u, K_b]$  is the probability that an adversary can infer  $u$ 's association with  $e_u$ .

Especially, the background knowledge  $K_b$  considered in this paper has stated in Section 3.2. To compute the identity linkability, it is sufficient to estimate the number of users contained in  $CR_u$ . During the cloaking process, we combine the noisy count and error analysis to guarantee generated  $CR_u$  containing enough users, i.e., the expected number of users exceeds than  $k$  with high probability. Further, it is easy to conclude that  $Pr[q \leftarrow u | CR_u, K_b] \leq \frac{1}{k}$ , which satisfies identity protection.

Following, we mainly analyze the location privacy guarantee in terms of location semantic. In order to evaluate the attack resilience of the location anonymization, we introduce a general *replay attack model*. In this model, for each segment  $e \in CR_u$ , by re-running the cloaking algorithm with  $e$  assumed to be the exact location, the adversary estimates the probability of  $e$  to generate  $CR_u$ ,  $like[CR_u | u \leftarrow e, K_b]$ . Then,  $Pr[u \leftarrow e_u | CR_u, K_b]$  is calculated as  $Pr[u \leftarrow e_u | CR_u, K_b] = \frac{like[CR_u | u \leftarrow e_u, K_b]}{\sum_{e \in CR_u} like[CR_u | u \leftarrow e, K_b]}$ .

According to the constructing principle of  $CR_u$ , it exists a subset  $CR_s \in CR_u$  which is firstly generated by semantic-based cloaking function. Also,  $\varphi(CR_s) \geq l$ . Besides, due to the existence of multiple locations sharing same semantics,  $|CR_s| \geq l$ . Based on *segment oblivious* property [29], for the query  $q$  issuing from each  $e$  in  $CR_s$  (i.e.,  $e \in CR_s$ ), the same  $CR_s$  can be generated. Thus, for any  $e$  in  $CR_s$ , it may be the query location. Further,  $Pr[u \leftarrow e_u | CR_u, K_b] \leq \frac{like[CR_u | u \leftarrow e_u, K_b]}{\sum_{e \in CR_s} like[CR_s | u \leftarrow e, K_b]} = \frac{like[CR_u | u \leftarrow e_u, K_b]}{(|CR_s| like[CR_u | u \leftarrow e_u, K_b])} \leq \frac{1}{l}$ . Alternatively, the semantic number of  $CR_s$  is no less than  $l$ , and hence it is difficult for an adversary to infer the semantic of query location. To sum up, our proposed cloaking algorithm can provide privacy protection for query users.

**Privacy guarantee for other users** During the location anonymization for query users, it requires to compute the user count for a particular segment. In our framework, PSD is used to release the noisy counts of the segments. This is the only step involving other users. If we can prove the released PSD does not violate location privacy for these users, then the whole algorithm will not leak their privacy. As stated above, in *PrivSem*,  $\epsilon$ -differential privacy is used to provide privacy protection for these users. Let  $D$  and  $D'$  denote two neighboring datasets. Therefore, our goal is to prove that  $\frac{Pr(A(D)=PSD_u)}{Pr(A(D')=PSD_u)} \leq e^\epsilon$ .

For our customized PSD,  $M_1$  be the sparse calculation mechanism,  $M_2$  be group mechanism, and  $M_3$  be the count publishing mechanism. Since changing a user location can affect the location count query is at most 2, and thus adding  $Lap(\frac{2}{\epsilon_1})$  satisfy  $\epsilon_1$ -differential privacy, i.e.,  $Pr(\mathcal{M}_1(D) = S) \leq e^{\epsilon_1} Pr(\mathcal{M}_1(D') = S)$ . Any postprocessing of differentially private data remains differentially private, i.e.,  $Pr(\mathcal{M}_2(\mathcal{M}_1(D) = S) = \mathfrak{R}) \leq e^{\epsilon_1} Pr(\mathcal{M}_2(\mathcal{M}_1(D') = S) = \mathfrak{R})$ . For a specified segment  $e$ , it will be either a independent

partition or a part of a partition. Based on this fact,  $Pr(\mathcal{A}(D, e) = \tilde{nc}_e)$  can be written as follows.

$$Pr(\mathcal{A}(D, e) = \tilde{nc}_e) = Pr(\mathcal{M}_2(\mathcal{M}_1(D, e) \geq \tau_1) = \mathfrak{R}_1)Pr(\mathcal{M}_3(D, e) = \tilde{nc}_e|\mathfrak{R}_1) + Pr(\mathcal{M}_2(\mathcal{M}_1(D, e) < \tau_1) = \mathfrak{R}_2)Pr(\mathcal{M}_3(D, e) = \tilde{nc}_e|\mathfrak{R}_2)$$

To analyze the expression above, we first investigate the properties of the following two expression:  $\frac{Pr(\mathcal{M}_3(D, e) = \tilde{nc}_e|\mathfrak{R}_1)}{Pr(\mathcal{M}_3(D', e) = \tilde{nc}_e|\mathfrak{R}_1)}$  and  $\frac{Pr(\mathcal{M}_3(D, e) = \tilde{nc}_e|\mathfrak{R}_2)}{Pr(\mathcal{M}_3(D', e) = \tilde{nc}_e|\mathfrak{R}_2)}$ .

$$\begin{aligned} \frac{Pr(\mathcal{M}_3(D, e) = \tilde{nc}_e|\mathfrak{R}_1)}{Pr(\mathcal{M}_3(D', e) = \tilde{nc}_e|\mathfrak{R}_1)} &\propto \frac{Pr(Lap(\tilde{nc}_e - nc_e^D))}{Pr(Lap(\tilde{nc}_e - nc_e^{D'}))} = \frac{\frac{1}{2\lambda} e^{-\frac{(\tilde{nc}_e - nc_e^D)}{\lambda}}}{\frac{1}{2\lambda} e^{-\frac{(\tilde{nc}_e - nc_e^{D'})}{\lambda}}} \\ &\leq e^{\frac{|nc_e^D - nc_e^{D'}|}{\lambda}} = e^{\epsilon^2} \end{aligned} \tag{8}$$

$$\begin{aligned} \frac{Pr(\mathcal{M}_3(D, e) = \tilde{nc}_e|\mathfrak{R}_2)}{Pr(\mathcal{M}_3(D', e) = \tilde{nc}_e|\mathfrak{R}_2)} &= \frac{Pr(\frac{\sum_{e_j \in gc} nc_{e_j}^D}{|gc|} + \frac{1}{|gc|} Lap(\frac{2}{\epsilon_2}) = \tilde{nc}_e)}{Pr(\frac{\sum_{e_j \in gc} nc_{e_j}^{D'}}{|gc|} + \frac{1}{|gc|} Lap(\frac{2}{\epsilon_2}) = \tilde{nc}_e)} \\ &\propto \frac{Pr(Lap(\frac{2}{\epsilon_2}) = (\tilde{nc}_e - \frac{\sum_{e_j \in gc} nc_{e_j}^D}{|gc|})|gc|)}{Pr(Lap(\frac{2}{\epsilon_2}) = (\tilde{nc}_e - \frac{\sum_{e_j \in gc} nc_{e_j}^{D'}}{|gc|})|gc|)} \\ &= \frac{\frac{1}{2\lambda} e^{-\frac{(\tilde{nc}_e - \frac{\sum_{e_j \in gc} nc_{e_j}^D}{|gc|})|gc|}{\lambda}}}{\frac{1}{2\lambda} e^{-\frac{(\tilde{nc}_e - \frac{\sum_{e_j \in gc} nc_{e_j}^{D'}}{|gc|})|gc|}{\lambda}}} \\ &\leq e^{\frac{\sum_{e_j \in gc} (|nc_{e_j}^{D'} - nc_{e_j}^D|)}{\lambda}} = e^{\frac{2}{\epsilon_2}} = e^{\epsilon^2} \end{aligned} \tag{9}$$

Combining the results of the (8, 9), we have

$$\begin{aligned} Pr(\mathcal{A}(D) = \tilde{nc}_e) &= Pr(\mathcal{M}_2(\mathcal{M}_1(D, e) \geq \tau_1) = \mathfrak{R}_1)Pr(\mathcal{M}_3(D, e) = \tilde{nc}_e|\mathfrak{R}_1) \\ &\quad + Pr(\mathcal{M}_2(\mathcal{M}_1(D, e) < \tau_1) = \mathfrak{R}_2)Pr(\mathcal{M}_3(D, e) = \tilde{nc}_e|\mathfrak{R}_2) \\ &\leq e^{\epsilon^1} Pr(\mathcal{M}_2(\mathcal{M}_1(D', e) \geq \tau_1) = \mathfrak{R}_1)e^{\epsilon^2} Pr(\mathcal{M}_3(D', e) = \tilde{nc}_e|\mathfrak{R}_1) \\ &\quad + e^{\epsilon^1} Pr(\mathcal{M}_2(\mathcal{M}_1(D', e) < \tau_1) = \mathfrak{R}_2)e^{\epsilon^2} Pr(\mathcal{M}_3(D', e) = \tilde{nc}_e|\mathfrak{R}_2) \\ &= e^{\epsilon^1 + \epsilon^2} (Pr(\mathcal{M}_2(\mathcal{M}_1(D', e) \geq \tau_1) = \mathfrak{R}_1)Pr(\mathcal{M}_3(D', e) = \tilde{nc}_e|\mathfrak{R}_1) \\ &\quad + Pr(\mathcal{M}_2(\mathcal{M}_1(D', e) < \tau_1) = \mathfrak{R}_2)Pr(\mathcal{M}_3(D', e) = \tilde{nc}_e|\mathfrak{R}_2)) \\ &= e^{\epsilon} Pr(\mathcal{A}(D', e) = \tilde{nc}_e) \end{aligned}$$

For the input  $D$ , we can utilize all segments to divide  $D$  into mutually disjoint sub-datasets. By the parallel composition property, the privacy budgets used in computing user count of each segment do not need to accumulate. Thus,  $Pr(\mathcal{A}(D, e) = PSD_u)$  is no more than  $e^{\epsilon} Pr(\mathcal{A}(D', e) = PSD_u)$ , which completes the proof.

**Table 1** Real dataset parameters

Name of dataset	Vertex count	Edge count	Semantic count	POIs count
Oldenburg (OL)	6,105	7,035	10	600
California (CA)	21048	21693	62	104,771

## 7 Experimental evaluation

In this section, we conduct extensive experiments to evaluate the effectiveness and efficiency of the proposed framework *PrivSem*. Our methods are implemented in C++. All the experiments are conducted on a machine with CPU Inter(R) Core(TM)i7-2600, 8.00GB memory, 3.40GHz frequency, 500GB hard disk.

### 7.1 Datasets and compared methods

- (1) **Datasets.** In the experiments, two real road network datasets are used, i.e., California and Oldenburg road networks<sup>1</sup>. These datasets involve diversified POIs, e.g., hospital, church, school, which we used as query objects in the experiments. The parameters of the two real road networks are summarized in Table 1.
- (2) **Query generator.** In all the experiments, we use the Network-based Generator of Moving Objects<sup>2</sup> to generate a set (10000) of moving objects on the maps. Because these two maps are of different scales, we can simulate both peak and off-peak traffic conditions. In each simulation, each moving object generates a set of (or none) KNN queries with a randomly assigned probability. The parameters of queries are listed in Table 2. More specifically, the parameters  $k$ ,  $l$ ,  $\sigma_t$ ,  $K$  and  $\gamma$  follow normal distribution. Particularly, after issuing a query request, the moving object waits for some inter-arrival time  $\gamma$ , until the request is either answered or dropped, before issuing another query request. The parameter  $c$  follows a uniform distribution over the interval  $[0, 62]$ . We consider privacy budget  $\epsilon \in \{0.5, 0.75, 1, 1.25, 1.5\}$ , ranging from strict to loose privacy requirements.
- (3) **Compared methods.** Under the *PrivSem* framework, we implement the following three methods: *EIRank*<sup>+</sup>, *Naive*, and *PrivSem*<sup>+</sup>. *EIRank*<sup>+</sup> is the algorithm which protects the identity and location privacy for query users over the road networks. In that case, we do not consider the other users' privacy and leverage the exact count of each segment. Conversely, the latter two methods take other users' privacy into consideration. The difference between them is the way to determine a CR: *Naive* determines an effective CR using naive user PSD, while *PrivSem*<sup>+</sup> uses the customized user PSD. In the experiments, we let  $\tau_1 = 15$ ,  $\tau_2 = 5$ ,  $\tau_3 = 0.5$ ,  $\alpha = 0.5$ , and  $\epsilon_1 = \frac{1}{2}\epsilon$  for all datasets. In addition, we compare our *EIRank*<sup>+</sup> method with *SA* algorithm [27]. This method uses  $k$ -anonymity and  $\theta$ -security semantics to protect query users on road networks. For  $\theta$ -security semantics, it is straightforward to convert it into  $l$ -semantic diversity.

<sup>1</sup><http://www.cs.utah.edu/~lifeifei/SpatialDataset.htm>

<sup>2</sup><http://www.fh-oow.de/institute/iapg/personen/brinkhoff>



**Table 2** Parameter setting

Parameters	$k$ -anonymity	$l$ -semantic diversity	$\sigma_t$	$\epsilon$ -DP	$c$	$KNN$	$\gamma$
Mean	5	3	10	N/A	N/A	5	20
Deviation	1.5	1.5	2	N/A	N/A	1	2

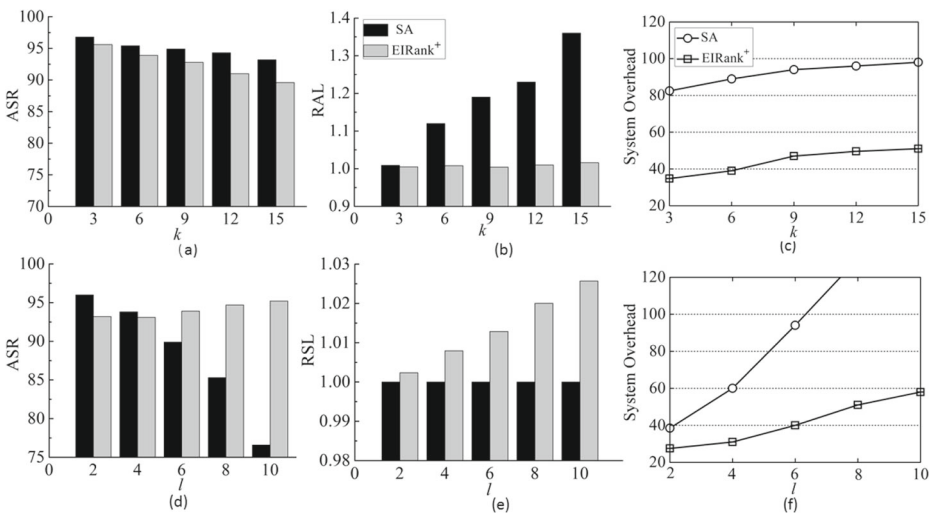
## 7.2 Experimental results

### A. Effectiveness of PrivSem for Query Users.

In the first set of experiments, we evaluate the utility of PrivSem framework for merely query users. Figure 7 shows the results with different parameters. From Figure 7a, it can be seen that ASR of both SA and  $EIRank^+$  tend to decrease as  $k$  increases. This is because, larger cloaking time required for anonymizing a query request for a larger  $k$ , resulting in many waiting requests drop. Figure 7b and c show that  $EIRank^+$  substantially outperforms SA in terms of RAL and system overhead. The main reason is that the cloaking strategies of the two algorithms are different.  $EIRank^+$  performs segment-based perturbation, which stops just after obtaining user specified requirement. In contrast, SA performs vertex Voronoi-based perturbation. Based on this difference, a CR of SA is larger than  $EIRank^+$ , and contains more users.

Figures 7d–f demonstrate the impact of varying semantic diversity on the performance for the two algorithms. From the results, we observe that with the increase of  $l$ ,  $EIRank^+$  consistently outperforms SA in terms of ASR and system overhead. What slightly surprised is, RSL of SA remains unchanged and that of  $EIRank^+$  increases. The phenomenon is reasonable. This is because the semantic number of a CR exactly equals to the user-defined semantic diversity for SA algorithm. To resist reverse engineering attacks, the latest CR of each bucket contains more than  $l$  semantics for  $EIRank^+$ .

Figure 8 depicts the time cost of two algorithms with different parameters. It is expected that when the location anonymization algorithm has to generate larger CRs



**Figure 7** Effectiveness comparison of  $EIRank^+$  and SA

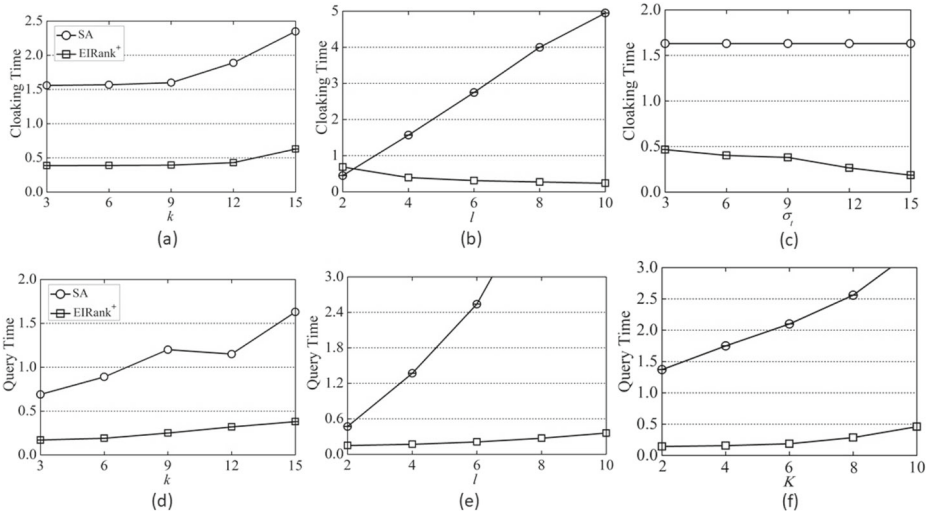


Figure 8 Time Cost of *EIRank+* and SA

to satisfy the stricter privacy requirements, the cloaking time of both approaches increases (Figure 8a). Such larger CRs lead to larger search space at the LBS provider, so the query quality gets worse when the privacy requirements become more stricter (Figure 8d and e). In addition, with the help of optimization strategies (RSC and DSA), the cloaking time of *EIRank+* decreases as  $l$  or  $\sigma_i$  becomes larger (Figure 8b and c). To retrieve more PoIs, i.e., a larger  $K$ , it can be seen that the query processing time increases.

B. Effect of Other Users' Privacy.

In this set of experiments, we mainly investigate the effect of considering others' user privacy. As shown in Figure 9, in general, *PrivSem+* is slightly inferior to *EIRank+* in terms of ASR, RAL(RSL), and system overhead.

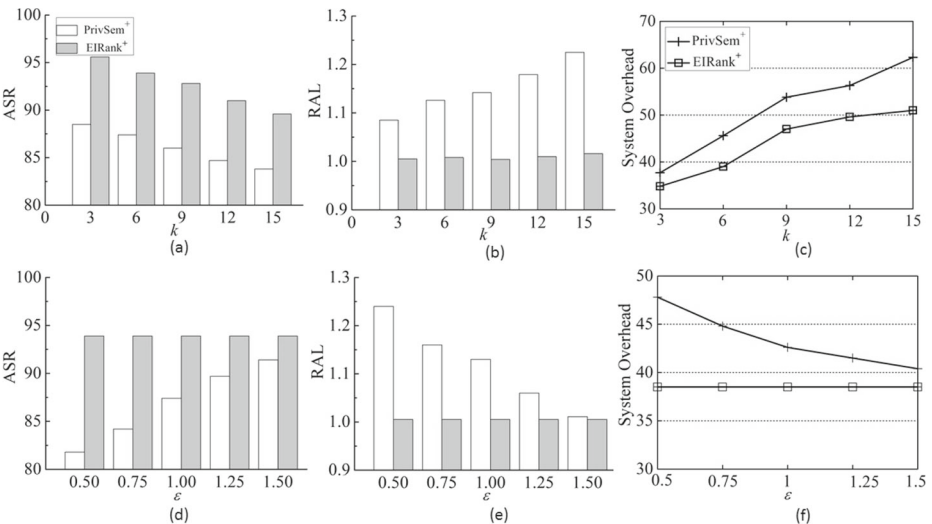
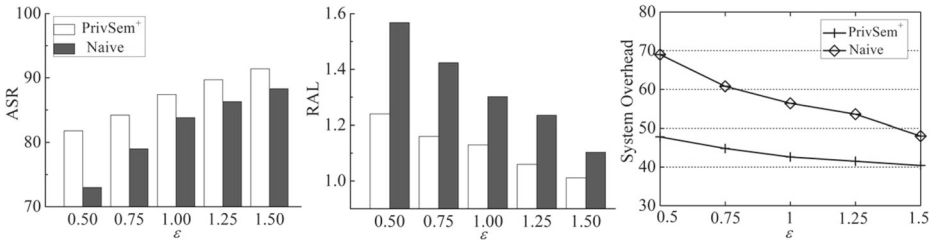


Figure 9 Effect of other users' privacy



**Figure 10** Effectiveness comparison of *PrivSem<sup>+</sup>* and *Naive*

**Varying  $k$ .** Figure 9a–c illustrate the performance of the two algorithms with different  $k$ . As shown, the performance of these two algorithms deteriorates as  $k$  increases. This is intuitive as a larger  $k$  imposes a stronger constraint on  $k$ -anonymity, thus taking more cloaking time and generating a larger CR. Further, for *PrivSem<sup>+</sup>*, a larger CR tends to introduce more noise. Compared with *EIRank<sup>+</sup>*, this noisy count brings more counting error. Hence, *PrivSem<sup>+</sup>* exhibits a relative poor performance.

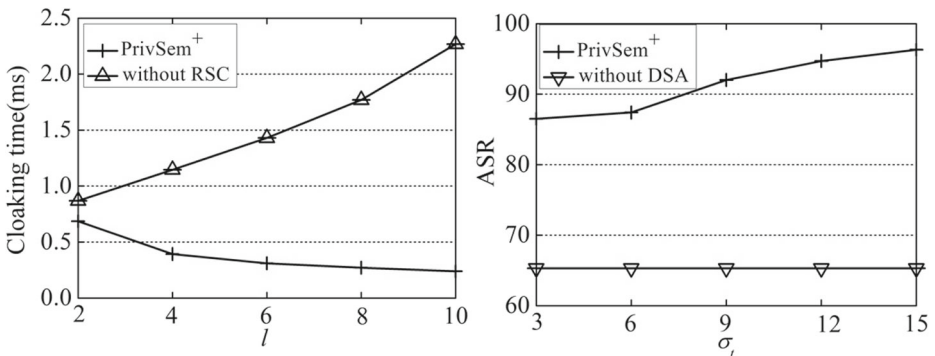
**Varying  $\epsilon$ .** Figure 9d–f show the performance of these two algorithms under varying privacy budget  $\epsilon$ . From the results, it is clearly that the performance of *EIRank<sup>+</sup>* is held constant. This is because, *EIRank<sup>+</sup>* is the exact algorithm, which is not affected by  $\epsilon$ . In addition, the utility of *PrivSem<sup>+</sup>* is improved as  $\epsilon$  increases. This conforms to the theoretical analysis that a larger privacy budget results in less noise and therefore a more accurate result.

C. *Effect of Customized User PSD.*

Figure 10 shows the performance of *Naive* and *PrivSem<sup>+</sup>* algorithms with different  $\epsilon$  values. Obviously, *PrivSem<sup>+</sup>* consistently gains better performance at the same level of privacy. Besides, all these two algorithms exhibit similar trend: the utility of the results is improved as  $\epsilon$  increases. This is because, when privacy budget  $\epsilon$  increases, a smaller amount of perturbation noise is required and a lower degree of privacy is guaranteed. Here, we omit the effect of  $k$  on these two algorithms since they present similar trend as Figure 9a–c.

D. *Effect of Optimization Strategies.*

In this part, we study how our optimization strategies affect the performance of *PrivSem<sup>+</sup>*. From Figure 11, we can see, without RSC, directly calculating the basic semantic-based cloaked area from the scratch each time produces poor results. It is



**Figure 11** Effect of optimization strategies

in line with our analysis: by effectively reducing the calculating number, the cloaking time of anonymizing requests can be significantly reduced. A similar trend also is observed. It means that our proposed optimization strategy DSA is effective in improving the utility of the cloaking algorithm.

## 8 Related work

In this section, we mainly discuss two main streams of research, location privacy and location semantics, which closely relate to our work. We introduce each stream in more details below.

### 8.1 Location privacy

In the past decades, with the explosive growth of LBS [23, 43], location privacy has received more and more attention [2, 3, 30, 38, 46]. Most techniques for achieving location privacy preservation can be categorized into *location anonymization* and *differential privacy based approaches*.

**Location anonymization** Location anonymization has gained popularity as a solution to preserve user location privacy in the LBS. It mainly leverages location obfuscation to perturb a user's exact location. Generally, it could be further classified into fake location [22, 40, 54], space transformation [6, 15, 36], mix-zones [35] and spatial cloaking. The main idea of false location is to send a fake location or a series of dummy locations including the user's exact location to the LBS providers. Its major shortcoming is expensive dummy generation cost. Sometimes, it cannot achieve claimed privacy protection level due to distance intersection attack [17]. Space transformation is a technique that transforms the original data space into another space while maintaining the spatial proximity, usually with cryptographic theory. Although the strong privacy provided, it also incurs significant computation cost that limits its applicability. The techniques based on mix-zones preserve a user's location by concealing his/her location in a zone. Among these diverse anonymization strategies, spatial cloaking is the prominent and the most relevant to ours.

Spatial cloaking blurs the exact location of a user with a CR until some privacy metrics are satisfied, such as  $k$ -anonymity [41] and  $l$ -diversity [31]. The location  $k$ -anonymity was initially by Gruteser et al. [16]. Subsequently, a series of research has been conducted to improve the computation of a CR. *CliqueCloak* [14] and *Casper* [34] both proposed personalized location anonymization. The former located a clique in a graph to compute the CR, while the latter utilized a quadtree-based index structure for fast computation of the CR. *HilbertCloak* [19] used Hilbert spacefilling curve and its CR is independent of mobile user distribution. Besides, there also exists other studies using other techniques to generate the CR, such as *Probabilistic Cloaking* [5], historical locations-based location privacy [50] and feeling-based location privacy [51].

Using location  $k$ -anonymity technique, the CR may include only one meaningful location (e.g., a specific clinic or church) and reveal strong relationships to such a location. To avoid this situation, *PrivacyGrid* [1] proposed location  $l$ -diversity, which enlarges a CR until ' $l-1$ ' different locations are included. Unfortunately, most of these existing spatial techniques are no longer applicable on road networks, because the area granularity of measurement tends to fail.

Recently, several studies have focused on location privacy preservation over road networks [7, 24, 25, 28, 45]. One of the best-known techniques is based on the concept of segment  $l$ -diversity [7, 45]. For instance, XSTAR [45] attempted to achieve the optimal balance between high query-processing efficiency and robust inference attack resilience while taking  $k$ -anonymity and segment  $l$ -diversity together into account. However, as mentioned, these techniques cannot prevent the location semantic information from leakage.

**Differential privacy based location privacy** Differential privacy is a strong privacy model which initiates in the statistics analysis, and then gradually are extended to location data. Until now, a lot of literature has focused on differentially private location data analysis and publishing. It is further divided into one-time release of static data [8, 20, 37, 47–49] and continuous release of dynamic data [12, 21, 44]. The work closely related to ours is differentially private location data publishing [8, 37, 48] for count queries. Generally, these studies resort to standard spatial indexing, e.g., grids and quad-trees, to provide a private description of the data distribution. Various fundamental steps, such as selecting splitting points and describing the data distribution within a region, must be done privately. In order to minimize the non-uniformly error, Xiao et al. [48] employed the heuristic to select the split points of KD-trees so that the two sub-regions are as close to uniform as possible. Instead of using a uniformity heuristic, Cormode et al. [8] split the nodes along the median of the partition dimension. Qardaji et al. [37] introduced a novel adaptive-grid method which lays a coarse-grained grid over the dataset, and then further partitions each cell according to its noisy count.

Differential privacy has also been applied to complex location data mining tasks. For instance, He et al. [18] studied differentially private trajectory publishing. To et al. [42] investigated location protection for worker datasets in spatial crowdsourcing. Li et al. [26] introduced private-preserving trajectory analysis for points-of-interest recommendation. These problems are orthogonal to ours.

## 8.2 Location semantics

Generally, the sensitive information is often exposed by query semantics or location semantics information. In the first case, it implies that the query content of a CR should be diverse. In this paper, our work focus on the second case, i.e., location semantics may disclose the sensitive information. For location  $l$ -diversity cloaking technique, the generated CR may embrace multiple locations. However, it may just include one place type. Several previous studies [9, 52] have identified such semantic breach issues.

Lee et al. [25] proposed mining the location semantic using Earth Mover's Distance to prevent location semantic from leakage. Yigitoglu et al. [53] extended the *semantic location cloaking model* [10] to provide location privacy protection in urban settings. Since the CRs are generated offline for a particular privacy requirement, it fails to support the varied privacy requirement. Recently, Li et al. [27] solved this issue based on the vertex Voronoi-partition. Unfortunately, similar to the work [25], it is vulnerable to *reverse engineering attacks*. To overcome these drawbacks, in the early time, we proposed EIRank [29] to resist semantic-based attack. The differences of this extended manuscript from our conference version [29] are as follows:

- We identify the privacy leakage problem of other users in the context of location-based queries, and present a framework that achieves differentially private guarantees, Section 3 summarizes this newly added content.

- We propose an error analysis model that quantifies the difference between noisy count of users in a cloaked area and its real count (Section 5.1), and we also devise a search strategy that find appropriate PSD regions to ensure high success rate of the cloaking algorithm (Section 5.2).
- We develop a series of optimization strategies to further improve the performance of the proposed framework. This newly added part in detailed in Section 5.3.
- Inspired by data sparsity issue, we design a customized user PSD to resist the influence of perturbed noise on counts, and Section 5.1 is newly added.
- We give the formal analysis about the privacy guarantee of *PrivSem* in newly added Section 6. Besides, we conduct more experiments on real datasets to evaluate its effectiveness and efficiency.

## 9 Conclusion

Protecting mobile user privacy is a fundamental problem in the LBS and has attracted intensive interests. However, most of these methods ignore semantic information and other users' privacy requirements. In this paper, we propose *PrivSem*, a novel framework which integrates location  $k$ -anonymity, segment  $l$ -semantic diversity and differential privacy to protect user privacy over road networks. Under this framework, we determine a CR using the sanitized data according to DP, instead of the original data. This task is challenging due to the uncertainty of DP. To address this, we present an error analysis model to quantify the error incurred in computing a CR. In addition, data sparsity in the spatial domain imposes another challenge to user privacy as well as utility. To address the issue, we further propose a customized user PSD which groups similar segments together to release more accurate data counts. Extensive experiments on several real road network datasets demonstrate the efficiency and effectiveness of our proposed framework *PrivSem*.

**Acknowledgments** This research was partially supported by the National Natural Science Foundation of China under Grant No. 61572119, 61622202, U1401256, 61732003 and 61729201; and the Fundamental Research Funds for the Central Universities under Grant No. N150402005.

## References

1. Bamba, B., Liu, L., Pesti, P., Wang, T.: Supporting anonymous location queries in mobile environments with privacygrid. In: Proceedings of WWW, pp. 237–246 (2008)
2. Beresford, A.R., Stajano, F.: Location privacy in pervasive computing. *Pervasive comput* **2**(1), 46–55 (2003)
3. Bettini, C., Mascetti, S., Wang, X.S.: Privacy Threats in Location-Based Services. In: Encyclopedia of GIS, pp. 906–912 (2008)
4. Chan, T.H.H., Shi, E., Song, D.: Private and continual release of statistics. *Information and System Security Journal* **14**(3), 26 (2011)
5. Cheng, R., Zhang, Y., Bertino, E., Prabhakar, S.: Preserving user location privacy in mobile data management infrastructures. *Lect. Notes Comput. Sci* **4258**, 393–412 (2006)
6. Chor, B., Goldreich, O., Kushilevitz, E., Sudan, M.: Private information retrieval. In: Proceedings of Annual Foundations of Computer Science, pp. 41–50 (1995)
7. Chow, C., Mokbel, M., Bao, J., Liu, X.: Query-aware location anonymization for road networks. *GeoInformatica* **15**(3), 571–607 (2011)
8. Cormode, G., Procopiuc, C., Srivastava, D., Shen, E., Yu, T.: Differentially private spatial decompositions. In: Proceedings of ICDE, pp. 20–31 (2012)

9. Damiani, M.L., Bertino, E., Silvestri, C., et al.: The probe framework for the personalized cloaking of private locations. *Trans. Data Privacy Journal* **3**(2), 123–148 (2010)
10. Damiani, M.L., Silvestri, C., Bertino, E.: Fine-grained cloaking of sensitive positions in location-sharing applications. *Pervasive Computing Journal* **10**(4), 64–72 (2011)
11. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: *Proceedings of TCC*, vol. 3876. pp. 265–284 (2006)
12. Fan, L., Xiong, L.: An adaptive approach to real-time aggregate monitoring with differential privacy. *TKDE J* **26**(9), 2094–2106 (2014)
13. Fogaras, D., Rácz, B.: A scalable randomized method to compute link-based similarity rank on the web graph. In: *Proceedings of EDBT Workshops*, pp. 557–567 (2004)
14. Gedik, B., Liu, L.: Location privacy in mobile systems: a personalized anonymization model. In: *Proceedings of ICDCS*, pp. 620–629 (2005)
15. Ghinita, G., Kalnis, P., Khoshgozaran, A., Shahabi, C., Tan, K.L.: Private queries in location based services: anonymizers are not necessary. In: *Proceedings of SIGMOD*, pp. 121–132 (2008)
16. Gruteser, M., Grunwald, D.: Anonymous usage of location-based services through spatial and temporal cloaking. In: *Proceedings of MobiSys*, pp. 31–42 (2003)
17. Hashem, T., Kulik, L., Ramamohanarao, K., Zhang, R., Soma, S.C.: Protecting privacy for distance and rank based group nearest neighbor queries. *World Wide Web* **22**(1), 375–416 (2019)
18. He, X., Cormode, G., Machanavajjhala, A., Procopiuc, C.M., Srivastava, D.: Dpt: differentially private trajectory synthesis using hierarchical reference systems. *VLDB J.* **8**(11), 1154–1165 (2015)
19. Kalnis, P., Ghinita, G., Mouratidis, K., Papadias, D.: Preventing location-based identity inference in an anonymous spatial queries. *TKDE J.* **19**(12), 1719–1733 (2007)
20. Kellaris, G., Papadopoulos, S.: Practical differential privacy via grouping and smoothing. In: *Proceedings of VLDB*, vol. 6. pp. 301–312 (2013)
21. Kellaris, G., Papadopoulos, S., Xiao, X., Papadias, D.: Differentially private event sequences over infinite streams. *VLDB J.* **7**(12), 1155–1166 (2014)
22. Kido, H., Yanagisawa, Y., Satoh, T.: An anonymous communication technique using dummies for location-based services. In: *Proceedings of ICPS*, pp. 88–97 (2005)
23. Kong, X., Song, X., Xia, F., Guo, H., Wang, J., Tolba, A.: Lotad: long-term traffic anomaly detection based on crowdsourced bus trajectory data. *World Wide Web* **21**(3), 825–847 (2018)
24. Ku, W.S., Zimmermann, R., Peng, W.C., Shroff, S.: Privacy protected query processing on spatial networks. In: *Proceedings of ICDE Workshop*, pp. 215–220. IEEE (2007)
25. Lee, B., Oh, J., Yu, H., Kim, J.: Protecting location privacy using location semantics. In: *Proceedings of SIGKDD*, pp. 1289–1297 (2011)
26. Li, C., Palanisamy, B.: Differentially private trajectory analysis for points-of-interest recommendation (2017)
27. Li, M., Qin, Z., Wang, C.: Sensitive semantics-aware personality cloaking on road-network environment. *International Journal of Security and Its Applications* **8**(1), 133–146 (2014)
28. Li, P., Peng, W., Wang, T.: A cloaking algorithm based on spatial networks for location privacy. In: *Proceedings of SUTC*, pp. 90–97 (2008)
29. Li, Y., Yuan, Y., Wang, G., Chen, L., Li, J.: Semantic-aware location privacy preservation on road networks. In: *Proceedings of DASFAA*, pp. 314–331 (2016)
30. Li, Z., Pei, Q., Liu, Y.: Spoofing attacks and countermeasures in fm indoor localization system. *World Wide Web* **21**(1), 219–240 (2018)
31. Machanavajjhala, A., Gehrke, J., Kifer, D.: Venkitasubramaniam, M.: l-diversity: Privacy beyond k-anonymity. In: *Proceedings of ICDE*, pp. 24–24 (2006)
32. Martínez-Hinarejos, C., Juan, A., Casacuberta, F.: Generalized k-medians clustering for strings. *Pattern Recognition and Image Analysis* pp. 502–509 (2003)
33. McSherry, F., Mironov, I.: Differentially private recommender systems: building privacy into the net. In: *Proceedings of SIGKDD*, pp. 627–636 (2009)
34. Mokbel, M.F., Chow, C.Y., Aref, W.G.: The new casper: Query processing for location services without compromising privacy. In: *Proceedings of VLDB*, pp. 763–774 (2006)
35. Palanisamy, B., Liu, L.: Mobimix: Protecting location privacy with mix-zones over road networks. In: *Proceedings of ICDE*, pp. 494–505 (2011)
36. Papadopoulos, S., Bakiras, S., Papadias, D.: Nearest neighbor search with strong location privacy. *VLDB J.* **3**(1-2), 619–629 (2010)
37. Qardaji, W., Yang, W., Li, N.: Differentially private grids for geospatial data. In: *Proceedings of ICDE*, pp. 757–768 (2013)
38. Shin, K.G., Ju, X., Chen, Z., Hu, X.: Privacy protection for users of location-based services. *Wirel. Commun. J.* **19**(1), 30–39 (2012)



39. Sibson, R.: Slink: an optimally efficient algorithm for the single-link cluster method. *Comput. J.* **16**(1), 30–34 (1973)
40. Soma, S.C., Hashem, T., Cheema, M.A., Samrose, S.: Trip planning queries with location privacy in spatial databases. *World Wide Web J.* **20**(2), 205–236 (2017)
41. Sweeney, L.: k-anonymity: A model for protecting privacy. *Int. J. Uncertainty Fuzziness Knowledge Based Syst.* **10**(05), 557–570 (2002)
42. To, H., Ghinita, G., Fan, L., Shahabi, C.: Differentially private location protection for worker datasets in spatial crowdsourcing. *TMC Journal* **16**(4), 934–949 (2017)
43. Vicente, C.R., Freni, D., Bettini, C., Jensen, C.S.: Location-related privacy in geo-social networks. *Internet Computing Journal* **15**(3), 20–27 (2011)
44. Wang, Q., Zhang, Y., Lu, X., Wang, Z., Qin, Z., Ren, K.: Rescuedp: Real-time spatio-temporal crowd-sourced data publishing with differential privacy. In: *Proceedings of INFOCOM*, pp. 1–9 (2016)
45. Wang, T., Liu, L.: Privacy-aware mobile services over road networks. *VLDB J.* **2**(1), 1042–1053 (2009)
46. Wu, W., Parampalli, U., Liu, J., Xian, M.: Privacy preserving k-nearest neighbor classification over encrypted database in outsourced cloud environments. *World Wide Web* pp. 1–23 (2018)
47. Xiao, X., Wang, G., Gehrke, J.: Differential privacy via wavelet transforms. *TKDE Journal* **23**(8), 1200–1214 (2011)
48. Xiao, Y., Xiong, L., Yuan, C.: Differentially private data release through multidimensional partitioning. *Secure Data Management Journal* **6358**, 150–168 (2010)
49. Xu, J., Zhang, Z., Xiao, X., Yang, Y., Yu, G., Winslett, M.: Differentially private histogram publication. *VLDB J.* **22**(6), 797–822 (2013)
50. Xu, T., Cai, Y.: Exploring historical location data for anonymity preservation in location-based services. In: *Proceedings of INFOCOM*, pp. 547–555 (2008)
51. Xu, T., Cai, Y.: Feeling-based location privacy protection for location-based services, In: *Proceedings of CCS*, pp. 348–357 (2009)
52. Xue, M., Kalnis, P., Pung, H.K.: Location diversity: Enhanced privacy protection in location based services. In: *Proceedings of LoCA*, pp. 70–87 (2009)
53. Yigitoglu, E., Damiani, M.L., Abul, O., Silvestri, C.: Privacy-preserving sharing of sensitive semantic locations under road-network constraints. In: *Proceedings of MDM*, pp. 186–195 (2012)
54. Yiu, M.L., Jensen, C.S., Huang, X., Lu, H.: Spacetwist: Managing the trade-offs among location privacy, query performance, and query accuracy in mobile services. In: *Proceedings of ICDE*, pp. 366–375 (2008)
55. Zhang, Y., Szabo, C., Sheng, Q.Z., Fang, X.S.: Snaf: Observation filtering and location inference for event monitoring on twitter. *World Wide Web* **21**(2), 311–343 (2018)

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.