Privacy of Location Trajectory

Chi-Yin Chow and Mohamed F. Mokbel

Abstract The ubiquity of mobile devices with global positioning functionality (e.g., GPS and Assisted GPS) and Internet connectivity (e.g., 3G and Wi-Fi) has resulted in widespread development of location-based services (LBS). Typical examples of LBS include local business search, e-marketing, social networking, and automotive traffic monitoring. Although LBS provide valuable services for mobile users, revealing their private locations to potentially untrusted LBS service providers pose privacy concerns. In general, there are two types of LBS, namely, snapshot and continuous LBS. For snapshot LBS, a mobile user only needs to report its current location to a service provider once to get its desired information. On the other hand, a mobile user has to report its location to a service provider in a periodic or on-demand manner to obtain its desired continuous LBS. Protecting user location privacy for continuous LBS is more challenging than snapshot LBS because adversaries may use the spatial and temporal correlations in the user's a sequence of location samples to infer the user's location information with a higher degree of certainty. Such user location trajectories are also very important for many applications, e.g., business analysis, city planning, and intelligent transportation. However, publishing original location trajectories to the public or a third party for data analysis could pose serious privacy concerns. Privacy protection in continuous LBS and trajectory data publication has increasingly drawn attention from the research community and industry. In this chapter, we describe the state-of-the-art privacy-preserving techniques for continuous LBS and trajectory publication.

Chi-Yin Chow
Department of Computer Science, City University of Hong Kong, Hong Kong, China, e-mail: chiychow@cityu.edu.hk

Mohamed F. Mokbel
Department of Computer Science and Engineering, University of Minnesota - Twin Cities, MN, USA, e-mail: mokbel@cs.umn.edu

1 Introduction

With the advanced location-detection technologies, e.g., global positioning system (GPS), cellular networks, Wi-Fi, and radio frequency identification (RFID), location-based services (LBS) have become ubiquitous [6, 30, 42]. Examples of LBS include local business search (e.g., searching for restaurants within a user-specified range distance from a user), e-marketing (e.g., sending e-coupons to nearby potential customers), social networking (e.g., a group of friends sharing their geo-tagged messages), automotive traffic monitoring (e.g., inferring traffic congestion from the position and speed information periodically reported from probe vehicles), and route finder applications (e.g., finding a route with the shortest driving time between two locations). There are two types of LBS, namely, *snapshot* and *continuous* LBS. For snapshot LBS, a mobile user only needs to report its current location to a service provider once to get its desired information. On the other hand, a mobile user has to report its location to a service provider in a periodic or on-demand manner to obtain its desired continuous LBS.

Although LBS provide many valuable and important services for end users, revealing personal location data to potentially untrustworthy service providers could pose privacy concerns. Two surveys reported in July 2010 found that more than half (55%) of LBS users show concern about their loss of location privacy [57] and 50% of U.S. residents who have a profile on a social networking site are concerned about their privacy [39]. The results of these surveys confirm that location privacy is one of the key obstacles for the success of location-dependent services. In fact, there are many real-life scenarios where perpetrators abuse location-detection technologies to gain access to private location information about victims [13, 15, 54, 55].

Privacy in continuous LBS is more challenging than snapshot LBS because adversaries could use the spatial and temporal correlations in the user's location samples to infer the user's private location information. Such user location trajectories are also very important for many real-life applications, e.g., business analysis, city planning, and intelligent transportation. However, publishing original location trajectories to the public or a third party for data analysis could pose serious privacy concerns. Privacy protection in continuous LBS and trajectory data publication has increasingly drawn attention from the industry and academia. In this chapter, we describe existing privacy-preserving techniques for continuous LBS and trajectory data publication.

The rest of this chapter is organized as follows. Section 2 presents the derivation of location trajectory privacy. Section 3 discusses the state-of-the-art privacy-preserving techniques for continuous LBS. Section 4 gives existing privacy protection techniques for trajectory publication. Finally, Section 5 concludes this chapter with research directions in privacy-preserving continuous LBS and trajectory data publication.

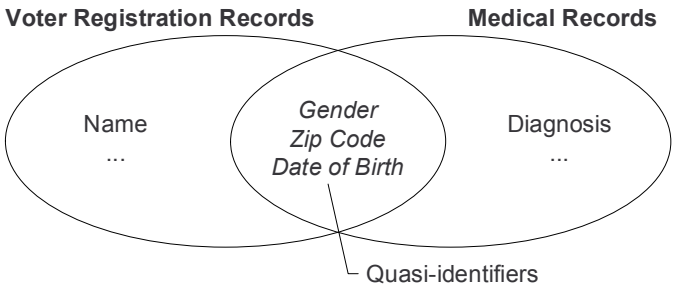


Fig. 1 Using quasi-identifiers to join two microdata sets.

2 The Derivation of Location Trajectory Privacy

This section gives the derivation of location trajectory privacy from data privacy and location privacy.

2.1 Data Privacy

Many agencies and other organizations often need to publish microdata, i.e., tables that contain unaggregated information about individuals, (e.g., medical, voter registration, census, and customer data) for many practical purposes such as demographic and public health research. In general, microdata are stored in a table where each row corresponds to one individual. In order to avoid the identification of records in microdata, known identifiers (e.g., name and social security number) must be removed. However, joining such *de-identified* microdata with other released microdata may still pose *data privacy* issues for individuals [49]. A study estimated that 87% of the population of the United States can be uniquely identified by using the collection of non-identity attributes, i.e., gender, date of birth, and 5-digit zip code [52]. In fact, those three attributes were used to link Massachusetts, USA voter registration records including name, gender, zip code and date of birth to *de-identified* medical data from Group Insurance Company including gender, zip code, date of birth and diagnosis to identify the medical records of the governor of Massachusetts in the medical data [52], as illustrated in Figure 1. Terminologically, attributes whose values taken together can potentially identify an individual record are referred to as *quasi-identifiers* and a set of records that have the same values for the quasi-identifiers in a released microdata is defined as an *equivalence class*.

Data privacy-preserving techniques are developed to anonymize microdata. Several data privacy principles are proposed to limit disclosure of anonymized microdata. For example, ***k*-anonymity** requires each record to be indistinguishable with other at least $k-1$ records with respect to the quasi-identifier, i.e., each equivalence class contains at least k records [35, 49, 51, 52]. However, a k-anonymized

equivalence class suffers from a homogeneity attack if all records in the class have less than k values for the sensitive attribute (e.g., disease and salary). To this end, **l-diversity** principle is proposed to ensure that an equivalence class must have at least l values for the sensitive attribute [38, 58]. To further strengthen data privacy protection, **t-closeness** principle is defined that an equivalence class is said to have t-closeness if the difference between the distribution of a sensitive attribute in this class and the distribution of the attribute in the entire data set is no more than a threshold parameter t [36]. For the details of these and other data privacy principles for data publishing, we refer the reader to the recent survey paper [18].

2.2 Location Privacy

In LBS, mobile users issue location-based queries to LBS service providers to obtain information based on their physical locations. LBS pose new challenges to traditional data privacy-preserving techniques due to two main reasons [41]. (1) These techniques preserve data privacy, but not the location-based queries issued by mobile users. (2) They ensure desired privacy guarantees for a snapshot of the database. In LBS, queries and data are continuously updated at high rates. Such highly dynamic behaviors need continuous maintenance of anonymized user and object sets.

Privacy-preserving techniques for LBS can be classified into three categories:

1. **False locations.** The basic idea of the techniques in this category is that the user sends either a fake location which is related to its actual location or its actual location with several fake locations to the LBS service provider in order to hide its location [28, 33, 62].
2. **Space transformation.** The techniques in this category transform the location information into another space where the spatial relationships among queries and data are encoded [20, 32].
3. **Spatial cloaking.** The main idea of spatial cloaking is to blur users' locations into cloaked spatial regions that are guaranteed to satisfy the k-anonymity [52] (i.e., the cloaked spatial region contains at least k users) and/or minimum region area privacy requirements [5, 14, 41] (i.e., the cloaked spatial region size is larger than a threshold) [2, 5, 7, 11, 12, 14, 19, 21, 22, 25, 31, 41, 64]. Spatial cloaking techniques have been extended to support road networks where a user's location is cloaked into a set of connected road segments so that the cloaked road segment set satisfies the privacy requirements of k-anonymity and/or minimum total road segment length [10, 34, 43, 56].

Anonymizing user locations is not the end of the story because database servers have to provide LBS based on anonymized user and/or object location information. Research efforts have also dedicated to dealing with privacy-preserving location-based queries, i.e., getting anonymous services from LBS service providers (e.g., [5, 20, 29, 31, 32, 41, 62]). These query processing frameworks can be divided into three main categories:

1. **Location obstruction.** The basic idea of location obstruction [62] is that a querying user first sends a query along with a fake location as an anchor to a database server. The database server keeps sending a list of nearest objects to the anchor to the user until the list of received objects satisfies the user's privacy and quality requirements.
2. **Space transformation.** This approach converts the original location of data and queries into another space. The space transformation maintains the spatial relationship among the data and query, in order to provide approximate query answers [20, 32] or exact query answers [20].
3. **Cloaked query area processing.** In this framework, a privacy-aware query processor is embedded into a database server to deal with the cloaked spatial region received either from a querying user [5, 29] or from a trusted third party [9, 31, 41]. For spatial cloaking in road networks, a query-aware algorithm is proposed to process privacy-aware location-based queries [3].

2.3 Trajectory Privacy

A location trajectory is a moving path or trace reported by a moving object in the geographical space. A location trajectory T is represented by a set of n time-ordered points, $T : p_1 \rightarrow p_2 \rightarrow \ldots \rightarrow p_n$, where each point p_i consists of a geospatial coordinate set (x_i, y_i) (which can be detected by a GPS-like device) and a timestamp t_i, i.e., $p_i = (x_i, y_i, t_i)$, where $1 \leq i \leq n$. Such spatial and temporal attributes of a location trajectory can be considered as powerful quasi-identifiers that can be linked to various other kinds of physical data objects [18, 44]. For example, a hospital releases a trajectory data set of its patients to a third-party research institute for analysis, as shown in Table 1. The released trajectory data set does not contain any explicit identifiers, such as patient name, but it contains a sensitive attribute (i.e., disease). Each record with a unique random ID, *RID*, corresponds to an individual, e.g., the record with $RID = 1$ means a patient visited locations $(1,5)$, $(6,7)$, $(8,10)$, and $(11,8)$ at timestamps 2, 4, 5, and 8, respectively. Suppose that an adversary knows that a patient of the hospital, Alice, visited locations $(1,5)$ and $(8,10)$ at timestamps 2 and 8, respectively. Since only the trajectory record with $RID = 1$ satisfies such spatial and temporal attributes, the adversary is 100% sure that Alice has HIV. This exam-

Table 1 Patient trajectory data.

RID	Trajectory	Disease	...
1	$(1,5,2) \rightarrow (6,7,4) \rightarrow (8,10,5) \rightarrow (11,8,8)$	HIV	...
2	$(5,6,1) \rightarrow (3,7,2) \rightarrow (1,5,6) \rightarrow (7,8,7) \rightarrow (1,11,8) \rightarrow (6,5,10)$	Flu	...
3	$(4,7,2) \rightarrow (4,6,3) \rightarrow (5,1,6) \rightarrow (11,8,8) \rightarrow (5,8,9)$	Flu	...
4	$(10,3,5) \rightarrow (7,3,7) \rightarrow (4,6,10)$	HIV	...
5	$(7,6,3) \rightarrow (6,7,4) \rightarrow (6,10,6) \rightarrow (4,6,9)$	Fever	...

ple shows that publishing *de-identified* user trajectory data can still cause serious privacy threats if the adversary has certain background knowledge.

In LBS, when a mobile user issues a continuous location-based query to a database server (e.g., "continuously inform me the traffic condition within 1 mile from my vehicle"), the user has to report its new location to the database server in a periodic or on-demand manner. Similarly, intelligent transportation systems require their users (e.g., probe vehicles) to periodically report their location and speed information to the system for analysis. Although such location-based queries and reports can be anonymized by replacing the identifiers of users with random identifiers, in order to achieve pseudonymity [47], the users may still suffer from privacy threats. This is because movements of whereabouts of users in public spaces can be openly observed by others through chance or engineered meetings [37]. In the worst case, if the starting location point of a trajectory is a home, an adversary uses reverse geocoding[1] [24] to translate a location point into a home address, and then uses a people-search-by-address engine (e.g., `http://www.intelius.com` and `http://www.peoplefinders.com`) to find the residents of the home address. Even though users generate a random identity for each of their location samples, multi-target tracking techniques (e.g., the multiple hypothesis tracking algorithm [48]) can be used to link anonymous location samples to construct target trajectories [26]. To this end, new techniques are developed to protect user location trajectory.

The key difference between continuous LBS and trajectory data publication with respect to challenges in privacy protection is twofold: (1) The scalability requirement of the privacy-preserving techniques for continuous LBS is much more important than that for trajectory data publication. This is because continuous LBS require the anonymization module to deal with a large number of real-time location updates at high rates while the anonymization process for trajectory data publication can be performed offline. (2) Global optimization can be applied to trajectory data publication because the anonymization process is able to analyze the entire (static) trajectory data to optimize its privacy protection or usability. However, global optimization is very difficult for continuous LBS, due to highly dynamic, uncertain user movements. Sections 3 and 4 present the state-of-the-art privacy-preserving techniques for continuous LBS and trajectory publication, respectively.

3 Protecting Trajectory Privacy in Location-based Services

In general, there are two categories of LBS based on whether they need a consistent user identity. A consistent user identity is not necessarily a user's actual identity or name because it can be an internal pseudonym.

[1] Reverse geocoding is the process of translating a human-readable address, such as a street address, from geographic coordinates, such as latitude and longitude.

- **Category-I LBS.** Some LBS require consistent user identities. For example, "Q1: let me know my friends' locations if they are within 2km from my location", "Q2: recommend 10 nearby restaurants to me based on my profile", and "Q3: continuously tell me the nearest shopping mall to my location". Q1 and Q2 require consistent user identities to let applications to find out their friends and profiles. Although Q3 does not need any consistent user identity, the query parameters (e.g., a particular shop name) can be considered as a virtual user identity that remains the same until the query expires.
- **Category-II LBS.** Other LBS do not require any consistent user identity, or even any user identity, such as "Q4: send e-coupons to users within 1km from my coffee shop".

In this section, we discuss seven privacy-preserving techniques for continuous LBS, including spatial cloaking, mix-zones, vehicular mix-zones, path confusion, path confusion with mobility prediction and data caching, Euler histogram-based on short IDs, and dummy trajectories, and indicate whether they support Category-I and/or II LBS from Sections 3.1 to 3.7, as summarized in Table 2.

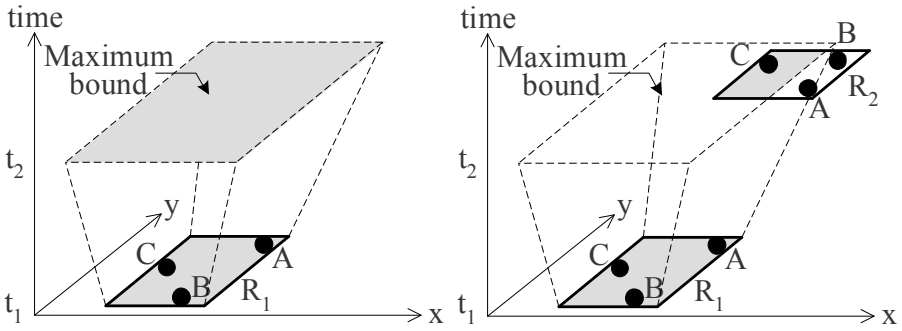
Table 2 Privacy-preserving techniques for continuous LBS.

Techniques	**Support Category-I LBS**	**Support Category-II LBS**
Spatial cloaking	Yes	Yes
Mix-zones	No	Yes
Vehicular mix-zones	No	Yes
Path confusion	No	Yes
Path confusion with mobility prediction and data caching	No	Yes
Euler histogram-based on short IDs	No	Yes
Dummy trajectories	Yes	Yes

3.1 Spatial Cloaking

Mobile users have to reveal their locations to database servers in a periodic or on-demand manner to obtain continuous LBS. Simply applying a snapshot spatial cloaking technique (e.g., [2, 14, 19, 21, 22, 25, 31, 41, 64]) to each user location independently cannot ensure k-anonymity for a user location trajectory. Specifically, we present two techniques, namely, *trajectory tracing* [5] and *anonymity-set tracing* [8], that can reduce the protection of spatial cloaking.

- **Trajectory tracing attack.** In case that an attacker can save the cloaked spatial regions of a querying user U and capture the maximum movement speed



(a) At time t_1 (b) At time t_2

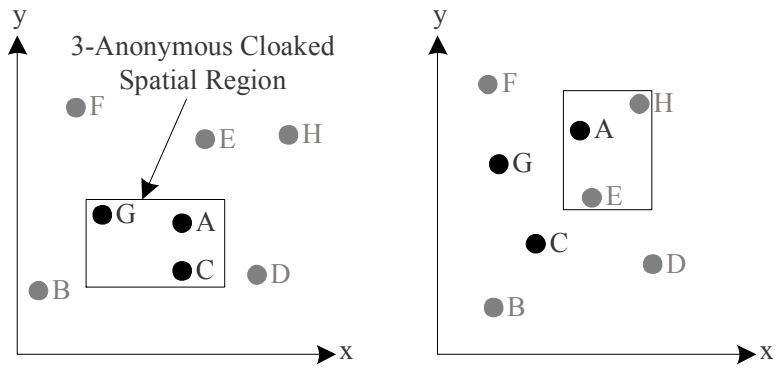
Fig. 2 Trajectory tracing attack.

max_{speed} of U based on U's historical movement patterns and/or other background information, e.g., the maximum legal driving speed and the maximum speed of U's vehicle, the attacker can use the trajectory tracing attack to reduce the effectiveness of spatial cloaking. Figure 2 depicts an example of the trajectory tracing attack, where the attacker collects a cloaked spatial region R_1 at time t_1 (represented by a solid rectangle in Figure 2a). The attacker cannot distinguish among the three users in R_1, i.e., A, B, and C. Figure 2b shows that the attacker collects another cloaked spatial region R_2 from U at time t_2. The attacker could use the most consecutive approach, where U moves at max_{speed} at any direction, to determine a *maximum bound* (represented by a dotted rectangle) that must contain U at time t_2. The maximum bound of R_1 at time t_2 can be determined by expanding each side of R_1 to a distance of $(t_2 - t_1) \times max_{speed}$. Since U must be inside R_2 and the maximum bound of R_1 at time t_2, the attacker knows that U is inside their overlapping area. Thus, the attacker knows that C is the querying user U.

- **Anonymity-set tracing attack.** An attacker could trace an anonymity set of a sequence of cloaked spatial regions of a continuous query to identify the query's sender [8]. Figure 3 gives an example of the anonymity-set tracing attack, where there are eight users A to H. The attacker collects two consecutive three-anonymous spatial regions at times t_1 and t_2, as depicted in Figures 3a and 3b, respectively. At time t_1, the probability of user A, C, or G being the query sender is 1/3. However, at time t_2, since only user A remains in the cloaked spatial region, the attacker knows that A is the query sender.

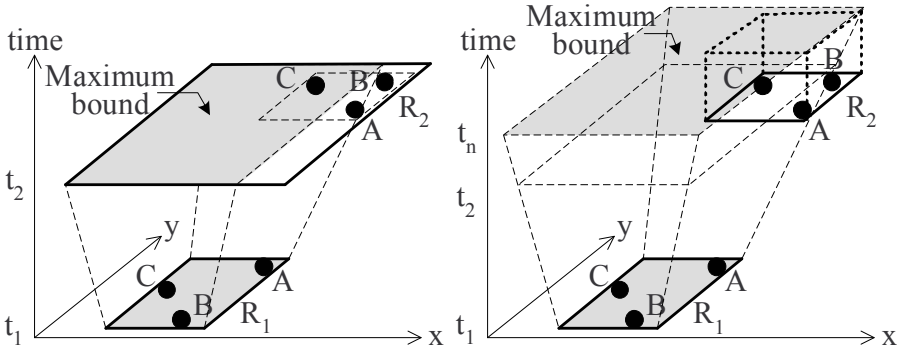
Patching and *delaying* techniques are proposed to prevent the trajectory tracing attack [5]. We describe these two techniques below.

- **Patching technique.** The patching technique is to combine the current cloaked spatial region and the maximum bound of the previous one such that the attacker can only know that the target user is inside the combined region. Figure 4a depicts



(a) At time t_1 (b) At time t_2

Fig. 3 Anonymity-set tracing attack.



(a) Patching technique (b) Delaying technique

Fig. 4 Two techniques for preventing the trajectory tracing attack.

an example for the patching technique in the running example, where the combination of the current cloaked spatial region at time t_2 (which is represented by a dotted rectangle) and the maximum bound of the cloaked spatial region released at time t_1 constitutes the user's cloaked spatial region R_2 (which is represented by a solid rectangle at time t_2).

- **Delaying technique.** The delaying technique is to suspend a location update until its cloaked spatial region is completely included in the maximum bound of the previous cloaked spatial region. As depicted in Figure 4b, the cloaked spatial region R_2 generated at time t_2 is suspended until R_2 fits into the maximum bound of the previous cloaked spatial region R_1 at time t_1. At time t_n, R_2 is reported to the database server.

In general, the patching technique generates larger cloaked spatial regions than the original ones, so it reduces the spatial accuracy of location updates that could

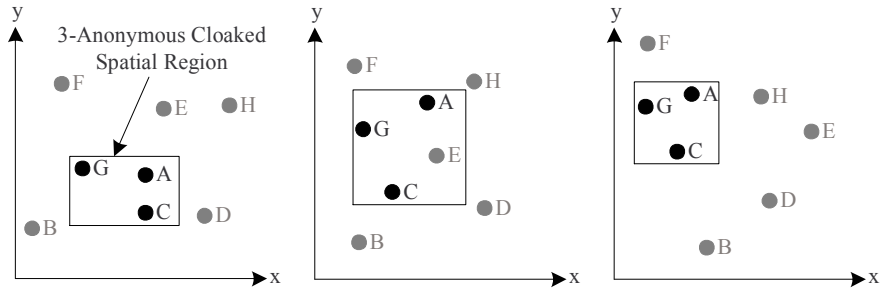


Fig. 5 Group-based spatial cloaking over real-time location trajectory data.

degrade the quality of services, in terms of the number of candidate answers returned to the user. On the other hand, the delaying technique does not reduce the spatial accuracy of location updates, but it degrades the quality of services in terms of the query response time.

To prevent the anonymity-set tracing attack, new spatial cloaking techniques based on either *real-time* or *historical* user trajectories are designed to protect user location trajectories. Similar to snapshot spatial cloaking techniques, a fully-trusted third party, usually termed *location anonymizer*, is placed between mobile users and database servers. The location anonymizer is responsible for collecting users' locations and blurring their locations into cloaked spatial regions that satisfy the user-specified k-anonymity level and/or minimum spatial region area. Since spatial cloaking techniques do not change user identities, they can support both Category I and II LBS.

In the following sections, we will discuss three main kinds of spatial cloaking techniques over user trajectories, namely, group-based, distortion-based, and prediction-based approaches, from Sections 3.1.1 to 3.1.3, respectively. All these techniques can prevent the anonymity-set tracing attack. The first two approaches are designed for real-time user trajectories, while the last one is for historical trajectory data.

3.1.1 Group-based Approach for Real-time Trajectory Data

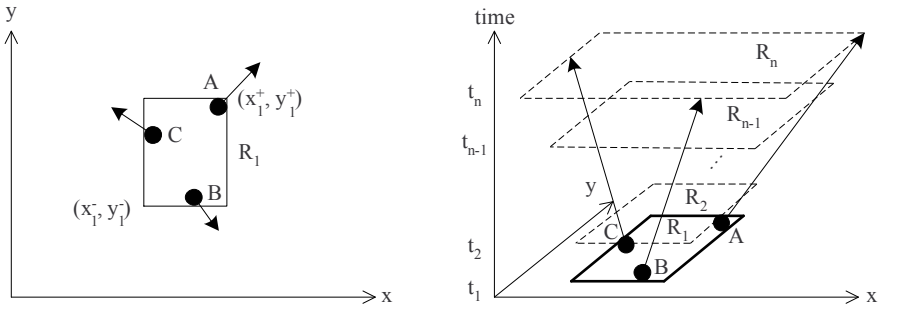
The group-based algorithm is proposed to use real-time location trajectory data to protect trajectory privacy for continuous location-based queries [8]. The basic idea is that a querying user U forms a group with other $k-1$ nearby peers. Before the algorithm issues U's location-based query or reports U's location to a database server, it blurs U's location into a spatial region that contains all the group members as a cloaked spatial region. Figure 5 depicts an example of continuous spatial cloaking over real-time user locations. In this example, user A that issues a continuous location-based query at time t_1 requires its location to be k-anonymized, where

$k = 3$. At time t_1, a location anonymizer forms a group of users A, C, and G, so that A's cloaked spatial region contains all these group members, as represented by a rectangle in Figure 5a. The location anonymizer sends A's query with its cloaked spatial region to the database server. At later times t_2 and t_3, when A reports its new location to the location anonymizer, a new cloaked spatial region that contains the group members is formed, as shown in Figures 5b and 5c, respectively. The drawbacks of this approach are that users not issuing any query have to report their locations to the location anonymizer and the cloaked spatial region would become very large after a long time period. Such a large cloaked spatial region may incur high computational overhead at the database server and result in many candidate answer objects returned from the database server to the location anonymizer.

In theory, let R_i be the cloaked spatial region for a querying user U at time t_i and $S(R_i)$ be a set of users located in R_i. Suppose U's query is first successfully cloaked at time t_1, it expires at time t_n, $U \in S(R_1)$, and $|S(R_1)| \geq k$. Without any additional information, the value of R_1's entropy, $H(R_1)$, is at least $\log|S(R_1)|$ which means that every user in R_1 has an equal chance of $1/|S(R_1)|$ to be U [60], i.e., R_1 is a k-anonymous cloaked spatial region for U. For U's cloaked spatial regions R_{i-1} and R_i generated at two consecutive times t_{i-1} and t_i $(1 < i \leq n)$, respectively, if R_{i-1} is a k-anonymous cloaked spatial region and $S(R_{i-1}) \subseteq S(R_i)$, R_i is also a k-anonymous cloaked spatial region [60]. Thus, the group-based approach can ensure k-anonymity for the entire life span of a continuous location-based query.

3.1.2 Distortion-based Approach for Real-time Trajectory Data

The distortion-based approach aims at overcoming the drawbacks of the group-based approach. It only requires querying users to report their locations to the location anonymizer, and it also considers their movement directions and velocities to minimize cloaked spatial regions [46]. A distortion function is defined to mea-



(a) Cloaked spatial region R_1 at time t_1

(b) Cloaked spatial regions $R_2, \ldots, R_n$ at times $t_2, \ldots, t_n$, respectively

Fig. 6 Query distortion for continuous spatial cloaking.

sure the temporal query distortion of a cluster of continuous queries. Figure 6 gives an example of how to determine query distortion. In this example, three users A, B and C that issue their continuous location-based queries at time t_1 constitute an anonymity set and their queries expire at time t_n. Their cloaked spatial region R_1 at time t_1 is a minimum bounding rectangle of the anonymity set, as represented by a rectangle (Figure 6a). Let (x_i^-, y_i^-) and (x_i^+, y_i^+) be the left-bottom and right-top vertices of a cloaked spatial region R_i at time t_i, respectively. The distortion for their queries with a cloaked spatial region R_i at time t_i is defined as:

$$\Delta(R_i) = \frac{(x_i^+ - x_i^-) + (y_i^+ - y_i^-)}{A_{height} + A_{width}}, \tag{1}$$

where A_{height} and A_{width} are the height and width of the minimum bounding rectangle of the entire system space, respectively. Assume their movement directions and velocities (represented by arrows in Figure 6b) remain the same from the time period t_1 to t_n, their subsequent cloaked spatial regions $R_2, R_3, \ldots, R_n$ at times $t_2, t_3, \ldots, t_n$ can be predicted, respectively. The distortion for their queries with respect to the time period from t_1 to t_n is defined as:

$$\int_{t_1}^{t_n} \Delta(R_i) = \frac{1}{A_{height} + A_{width}} \left\{ \int_{t_1}^{t_2} \Delta(R_1)dt + \int_{t_2}^{t_3} \Delta(R_2)dt + \ldots + \int_{t_{n-1}}^{t_n} \Delta(R_n)dt \right\},$$

Given a new continuous location-based query Q, greedy cloaking and bottom-up cloaking algorithms are designed to cluster Q with other $k-1$ outstanding queries into a group such that the group satisfies k-anonymity and their query distortion is minimized [46].

3.1.3 Predication-based Approach for Historical Trajectory Data

Another way to ensure k-anonymity is to use individuals' historical footprints, instead of their real-time locations [61]. A footprint is defined as a user's location col-

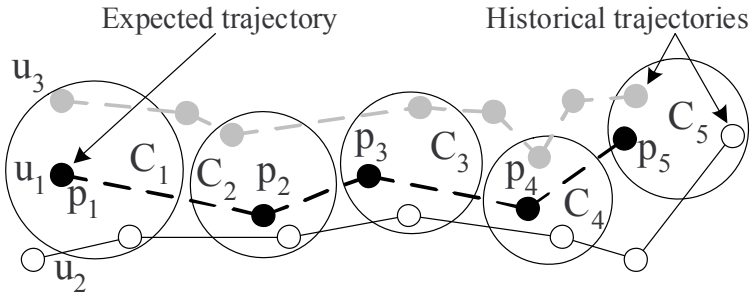


Fig. 7 Continuous spatial cloaking over historical trajectories.

lected at some point of time. Similar to the previous two approaches, a fully-trusted location anonymizer is placed between users and LBS service providers to collect users' footprints. Given a user's predicted trajectory (i.e., a sequence of expected footprints), the location anonymizer cloaks it with $k-1$ historical trajectories collected from other users. Figure 7 gives an example for continuous spatial cloaking over historical trajectories, where a user u_1 wants to subscribe continuous LBS from a service provider. u_1's predicted time-ordered footprints, $p_1 \rightarrow p_2 \rightarrow \ldots \rightarrow p_5$, are represented by black circles. If u_1's desired anonymity level is $k = 3$, the location anonymizer finds historical trajectories from other two users, u_2 and u_3. Then, each u_1's expected footprint p_i ($1 \leq i \leq 5$) is cloaked with at least one unique footprint of each of u_2's and u_3's trajectories to form a cloaked spatial region C_i. The sequence of such cloaked spatial regions constitutes the k-anonymized trajectory for u_1.

Given a k-anonymized trajectory $T = \{C_1, C_2, \ldots, C_n\}$, its resolution is defined as:

$$|T| = \frac{\sum_{i=1}^{n} Area(C_i)}{n}, \tag{2}$$

where $Area(C_i)$ is the area of cloaked spatial region C_i. For quality of services, $|T|$ should be minimized. Since the computation of an optimal T would be expensive, heuristic approaches are designed to find T [61]. Although using historical trajectory data gives better resolutions for k-anonymized trajectories, it would suffer from an observation attack. This is because an attacker may only see the querying user or less than k users located in a cloaked spatial region at its associated timestamp.

3.2 Mix-Zones

The concept of "mix" has been applied to anonymous communication in a network. A mix-network consists of normal message routers and mix-routers. The basic idea is that a mix-router collects k equal-length packets as input and reorders them randomly before forwarding them, thus ensuring unlinkability between incoming and outgoing messages. This concept has been extended to LBS, namely, *mix-zones* [4]. When users enter a mix-zone, they change to a new, unused pseudonym. In addition, they do not send their location information to any location-based application when they are in the mix-zone. When an adversary that sees a user U exits from the mix-zone cannot distinguish U from any other user who was in the mix-zone with U at the same time. The adversary is also unable to link people entering the mix-zone with those coming out of it. A set of users S is said to be k-anonymized in a mix-zone Z if all following conditions are met [45]:

1. The user set S contains at least k users, i.e., $|S| \geq k$.
2. All users in S are in Z at a point in time, i.e., all users in S must enter Z before any user in S exits.
3. Each user in S spends a completely random duration of time inside Z.

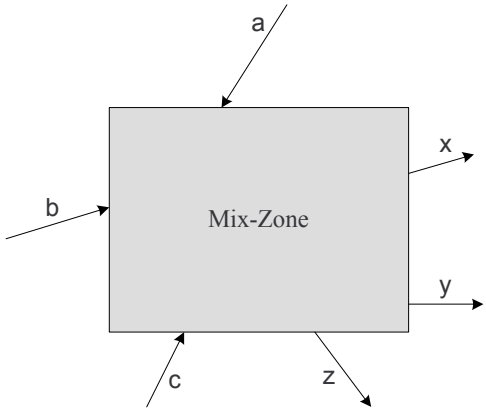


Fig. 8 A mix-zone with three users.

Table 3 An example of three-anonymized mix-zone.

User ID	**Old Pseudonyms**	**New Pseudonyms**	$time_{enter}$	$time_{exit}$	$time_{inside}$
α	a	y	2	9	7
β	c	x	5	8	3
γ	b	z	1	11	10

4. The probability of every user in S entering through an entry point is equally likely to exit in any of the exit points.

Table 3 gives an example of three-anonymity for the mix-zone depicted in Figure 8, where three users with real identities, α, β, and γ enter the mix-zone with old pseudonyms a, c, and b at times ($time_{enter}$) 2, 5, and 1, respectively. Users α, β, and γ exit the mix-zone with new pseudonyms y, x, and z at times ($time_{exit}$) 9, 8, and 11, respectively. Thus, they all are in the mix-zone during the time period from 5 to 8. Since they stay inside the mix-zone with random time periods (i.e., $time_{inside}$), there is a strong unlinkability between their entering order ($\gamma \rightarrow \alpha \rightarrow \beta$) and exiting order ($\beta \rightarrow \alpha \rightarrow \gamma$).

We can see that mix-zones require pseudonym change to protect user location privacy, so this technique can only support Category-II LBS. Mix-zones also impose limits on the services available to mobile users inside a mix-zone because they cannot update their locations until exiting the mix-zone. To minimize disruptions caused to users, the placement of mix-zones in the system should be optimized to limit the total number of mix-zones required to achieve a certain degree of anonymity [17].

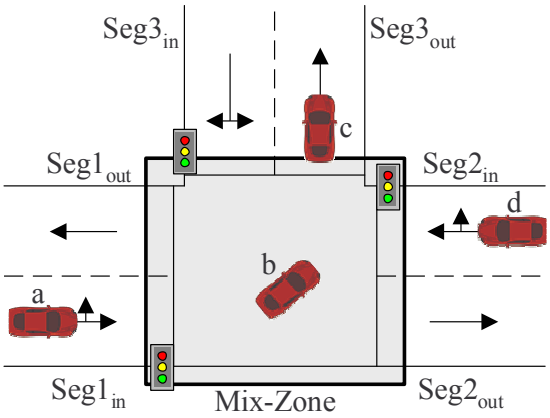


Fig. 9 A vehicular mix-zone.

3.3 Vehicular Mix-Zones

In a road network, vehicle movements are constrained by many spatial and temporal factors, such as physical roads, directions, speed limits, traffic conditions, and road conditions. Mix-zones designed for the Euclidean space are not secure enough to protect trajectory privacy in road networks [16, 45]. This is because an adversary can gain more background information from physical road constraints and delay characteristics to link entering events and exiting events of a mix-zone with a high degree of certainty. For example, a mix-zone (represented by a shaded area) is placed on an intersection of three road segments $Seg1$, $Seg2$, and $Seg3$, as depicted in Figure 9. If u-turn is not allowed in the intersection, an adversary knows that a vehicle with pseudonym c enters the mix-zone from either $Seg1_{in}$ or $Seg2_{in}$. Since a vehicle turning from $Seg1_{in}$ to $Seg3_{out}$ normally takes a longer time than turning from $Seg2_{in}$ to $Seg3_{out}$, the adversary would use this delay characteristic to link an exiting event at $Seg3_{out}$ to an entering event at $Seg1_{in}$ or $Seg2_{in}$. In addition, every vehicle may spend almost the same time during a short time period for a specific direction, e.g., u-turn, left, straight, or right. This temporal characteristic may violate the third necessary condition for mix-zones listed in Section 3.2.

An effective solution for vehicular mix-zones is to construct non-rectangular, adaptive mix-zones that start from the center of an road segment intersection on its outgoing road segments [45], as depicted in Figure 10. The length of each mix-zone on an outgoing segment is determined based on the average speed of the road segment, the time window, and the minimum pairwise entropy threshold. The dark shaded area should also be included in the mix-zone to ensure that an adversary cannot infer the vehicle movement direction (e.g., turn left or go straight in this example). The pairwise entropy is computed for every pair of users a and b in an anonymity set S by considering a and b to be the only members in S and determining the linkability between their old and new pseudonyms. Similar to mix-zones, vehic-

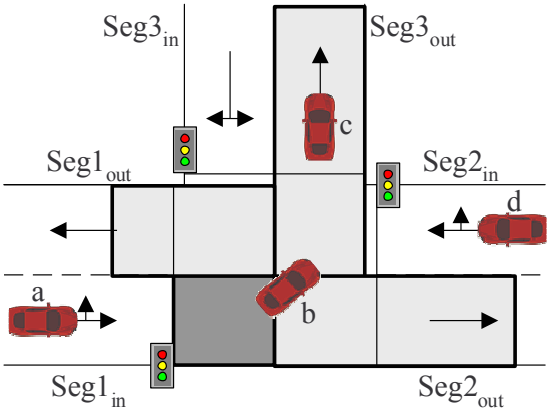


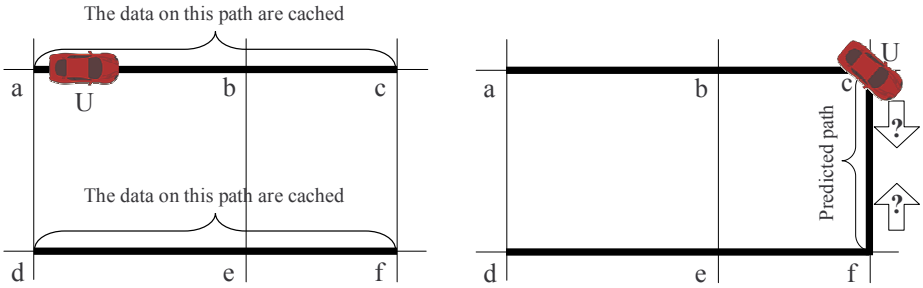
Fig. 10 Non-rectangular, adaptive vehicular mix-zones.

ular mix-zones require a pseudonym change, so they can only support Category-II LBS.

3.4 Path Confusion

Since consecutive location samples from a vehicle are temporally and spatially correlated, trajectories of individual vehicles can be constructed from a set of location samples with anonymized pseudonyms reported from several vehicles through target tracking algorithms [26]. The general idea of these algorithms is to predict the position of a target vehicle based on the last known speed and direction information and then decide which next location sample (or the one with the highest probability if there are multiple candidate location samples) to link to the same vehicle through Maximum Likelihood Detection [26].

The main goal of the path confusion technique is to avoid linking consecutive location samples to individual vehicles through target tracking algorithms with high confidence [27]. The degree of privacy of the path confusion technique is defined as the "time-to-confusion", i.e., the tracking time between two location samples where an adversary could not determine the next sample with sufficient tracking certainty. Tracking uncertainty is computed by $H = -\sum p_i \log p_i$, where p_i is the probability that location sample i belongs to a target vehicle. Smaller values of H means higher certainty or lower privacy. Given a maximum allowable time to confusion, *ConfusionTime*, and an associated uncertainty threshold, *ConfusionLevel*, a vehicle's location sample can be safely revealed if the time between the current time t and the last point of its confusion is less than *ConfusionTime* and the tracking uncertainty of its sample with all other location samples revealed at time t is higher than *ConfusionLevel*. To reduce computational overhead, the computation of tracking uncertainty can only consider the k-nearest location samples to a predicted location point (cal-



(a) Using cached data (b) Predicting a movement path

Fig. 11 Path confusion with mobility prediction and data caching.

culated by the target tracking algorithm), rather than all location samples reported at time t. Since a consistent user identity can link consecutive location samples, it cannot be revealed to any service provider; and hence, the path confusion technique only supports Category-II LBS.

3.5 Path Confusion with Mobility Prediction and Data Caching

The main idea of the path confusion technique with mobility prediction and data caching, called *CacheCloak*, is that the location anonymizer predicts vehicular movement paths, pre-fetches the spatial data on predicted paths, stores the data in a cache [40]. Figure 11a illustrates an example for *CacheCloak* where there are seven road segments, ab, bc, de, ef, ad, be, and cf, with six intersections, a to f. A bold road segment indicates that the data located on it are currently cached by the location anonymizer (e.g., data on road segments ab, bc, de, and ef are cached). Given a continuous location-based query from a user U (represented by a car in Figure 11), if U is moving on a path whose data are currently cached by the location anonymizer, the location anonymizer is able to return the query answer to U without contacting any LBS service provider, as depicted in Figure 11a. On the other hand, Figure 11b shows that U enters a road segment (i.e., cf) from an intersection c with no data in the cache. In this case, the location anonymizer predicts a path (which contains one road segment cf) for U that connects to an intersection of existing paths with cached data (i.e., f). Then, the location anonymizer issues a query Q to the service provider to get the data on the newly predicted path cf and caches the data. It finally returns the answer of Q to U.

CacheCloak prevents the LBS service provider from tracking any one user because the service provider can only see queries for a series of interweaving paths [40]. As shown in the example, the service provider cannot track the user who issues Q because the user could be turning in from road segment bc or from the other one ef (Figure 11b). *CacheCloak* can tolerate mispredictions or dynamic

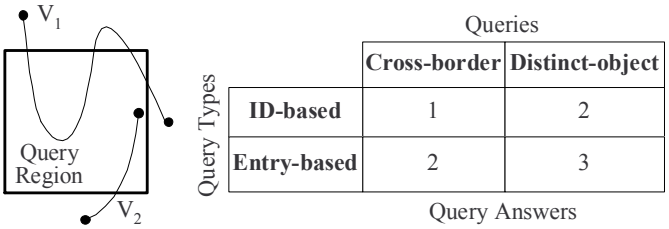


Fig. 12 ID- and entry-based aggregate queries in road networks.

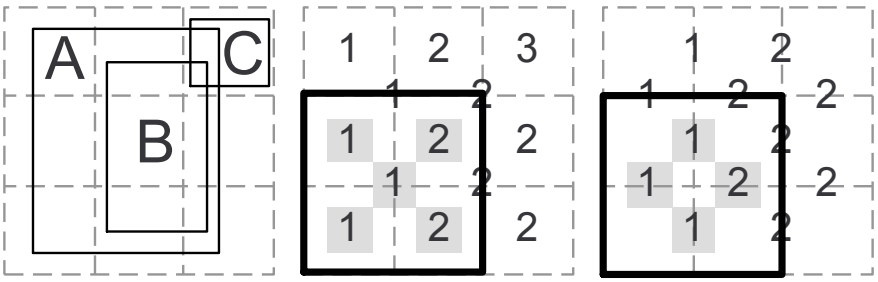
data. When a user deviates from a predicted path p or the cached data for p have expired, the location anonymizer simply makes a new prediction. To preserve wireless bandwidth, stale data or data on mispredicted road segments will not be sent to the user. *CacheCloak* is a variant version of the basic path confusion technique, as described in Section 3.4, so it also only supports Category-II LBS.

3.6 Euler Histogram-based on Short IDs

The Euler Histogram-based on Short IDs (EHSID for short) is proposed for providing privacy-aware traffic monitoring services through answering aggregate queries [59]. EHSID supports two types of aggregate queries in road networks, namely, *ID-* and *entry-based* query types.

ID-based query type. ID-based queries ask for the count of unique vehicles inside a query region in a road network. Given a query region R, a *cross-border* query collects the count of unique vehicles that cross the boundary of R and a *distinct-object* query collects the count of unique vehicles that have been detected in R, including R's boundary. Figure 12 gives an example with two vehicles v_1 and v_2 whose trajectories intersect a rectangular query region R. Since there is only one vehicle, i.e., v_1, crossing the boundary of R, the *cross-border* query answer is one. On the other hand, both v_1 and v_2 are in R, so the *distinct-object* query answer is two.

Entry-based query type. Entry-based queries ask for the number of entries to a query region. If a vehicle has multiple entries to a query region (i.e., it enters the query region several times), its trajectory is divided into multiple sub-trajectories. Entry-based queries count the number of entries to the query region, even if the entries are from the same vehicle. Given a query region R, a *cross-border* query counts the number of entries that cross the boundary of R and a *distinct-object* query counts the number of distinct entries to R, including its boundary. Figure 12 depicts that v_1's trajectory can be divided into two sub-trajectories and each one crosses the boundary of the query region R, so the *cross-border* query answer is two. With an additional entry from v_2, the *distinct-object* query answer is three.



(a) 3×3 grid with 3 rectangles (b) Counts for faces and vertices (c) Counts for edges

Fig. 13 Using an Euler histogram to count distinct rectangles in a query region.

The basic idea of EHSID is based on Euler histograms, which were designed to count the number of rectangular objects in multidimensional space [50]. An Euler histogram is constructed by partitioning the space into a grid and maintaining a count for the number of intersecting objects of each face, edge, and vertex in the grid. Figure 13a gives an example of a 3×3 gird with three rectangular objects A, B, and C. Figure 13b shows the count of each face and vertex in the grid, and Figure 13c shows the count of each edge in the grid. Given a query region R, which is represented by a bold rectangle, the total number of distinct objects N in R is estimated by the equation $N = F + V - E$, where F is the sum of face counts inside R, V is the sum of vertex counts inside R (excluding its boundary), and E is the sum of edge counts inside R (excluding its boundary). In this example, $F = 1+2+1+2 = 6$, $V = 1$, and $E = 1+1+1+2 = 5$; hence, $N = 6+1-5 = 2$, which is the exact number of distinct rectangles intersecting R.

To protect user trajectory privacy, each vehicle is identified by a short ID that is extracted from random bit positions from its full ID and EHSID periodically changes the bit positions. For example, if a vehicle's full ID is "$\underline{1}10\underline{1}11\underline{0}11$" and a bit pattern is 1, 3, 4, and 7, its short ID generated by the system is "1010". The random bit pattern is periodically updated. Let l_F and l_S be the length of the full and short IDs, respectively. The degree of k-anonymity of a short ID is measured as $k = 2^{l_F - l_S}$ [59].

Figure 14 gives a simple example with two vehicles for EHSID, where the road network consists of six vertices a to f and five road segments, ab, bc, cd, de, and ef. The short IDs of the two vehicles are "01" and "10". EHSID maintains a data list for each vertex and edge. An item in a data list is defined as (*short ID*, *the counter for the short ID*). The query region of a query Q is shown as a bold rectangle. In general, the query processing algorithm of EHSID has two main steps.

1. **Aggregating data at vertices and edges.** The algorithm identifies four types of relevant vertices to a query, as depicted in Table 4. An relevant edge is denoted as $E_{x,y}$, where x is the vertex type at one end and y is the vertex type at the other end. For example, $E_{jo,ob}$ denotes an edge with a *just-outside* vertex (V_{jo}) at one end

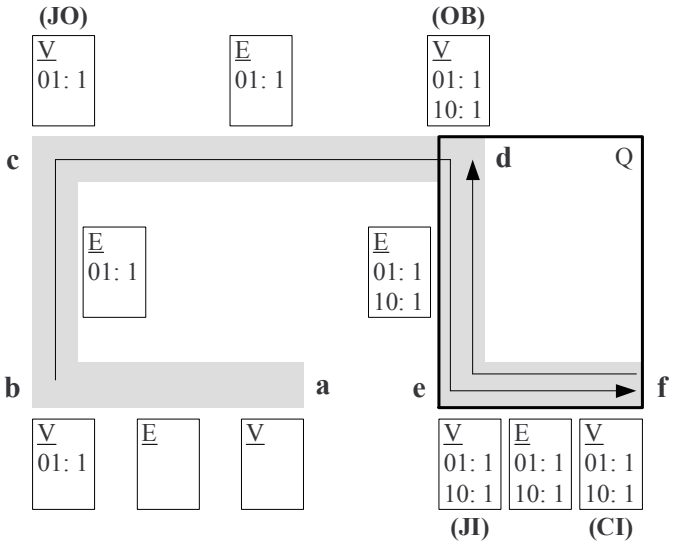


Fig. 14 EHSID with two trajectories.

Table 4 Four types of vertices in EHSID.

Vertex type	**Relative position to query region R**
Just Outside (V_{jo})	Located outside R and has a one-hop neighbor inside R
On Border (V_{ob})	Located inside R and has a one-hop neighbor outside R
Just Inside (V_{ji})	Located inside R and no one-hop neighbor is V_{jo} but has V_{ob} as an one-hop neighbor
Completely Inside (V_{ci})	Located inside R but is not V_{ob} or V_{ji}

Table 5 Relevant short IDs and edge datasets for a query.

Query type	**Relevant short IDs**	**Relevant edge datasets**
ID-based cross-border	$V_{jo} \cap V_{ob} \cap V_{ji}$	N/A
ID-based distinct-objects	$V_{ob} \cup V_{ji} \cup V_{ci}$	N/A
Entry-based cross-border	$V_{jo} \cap V_{ob} \cap V_{ji}$	$E_{jo,jo}$, $E_{ob,ob}$, $E_{ji,ji}$, $E_{jo,ob}$, $E_{ob,ji}$
Entry-based distinct-objects	$V_{ob} \cup V_{ji} \cup V_{ci}$	$E_{ob,ob}$, $E_{ji,ji}$, $E_{ci,ci}$, $E_{ob,ji}$, $E_{ji,ci}$

and an *on-border* vertex (V_{ob}) at the other end. Then, the algorithm aggregates the data for each type of vertices and edges. If a short ID appears n times ($n > 0$) for the same vertex or edge type, its corresponding counter is simply set to n.

2. **Computing query answers.** Since a short ID of a vehicle may not be relevant to a query, the algorithm needs to find a set of relevant short IDs based on the definition of the query. The relevant set of short IDs and aggregate edge datasets to a query are defined in Table 5. EHSID only needs the relevant short IDs to compute answers for ID-based aggregate queries. If the query Q de-

Table 6 Privacy measures of the example in Figure 15.

Time (i)	t_1	t_2	t_3	t_4	t_5
Real trajectory (T_r)	(1,2)	(2,3)	(3,3)	(4,3)	(5,3)
Dummy (T_{d_1})	(1,4)	(2,3)	(2,2)	(2,1)	(3,1)
Dummy (T_{d_2})	(4,4)	(3,4)	(3,3)	(3,2)	(4,2)
$\|S_i\|$	3	2	2	3	3
$\frac{1}{n}\sum_{j=1}^{n} dist(T_r^i, T_{d_j}^i)$	2.80	0.71	0.71	2.12	2.12

picted in Figure 14 is an ID-based cross-border query, the relevant short ID to Q is $V_{jo} \cap V_{ob} \cap V_{ji} = \{01\} \cap \{01,10\} \cap \{01,10\} = \{01\}$. Since there is only one relevant ID, the query answer is one. On the other hand, if Q is an ID-based distinct-objects query, the relevant short IDs to Q are $V_{ob} \cup V_{ji} \cup V_{ci} = \{01,10\} \cup \{01,10\} \cup \{01,10\} = \{01,10\}$. Since there are two relevant IDs, the query answer is two. In this example, EHSID finds the exact query answers for both the ID-based cross-border and distinct-objects queries.

EHSID uses a different method to compute answers for entry-based queries. After finding the relevant short IDs, for each relevant ID, we find the total count C_v for the ID in all the aggregate vertices (which are computed by the first step) and the total count C_e for the ID in all the relevant aggregate edge datasets. The query answer is computed as $C_v - C_e$. If Q is an entry-based cross-border query, there is only relevant ID, $\{01\}$. $\{01\}$ appears once in V_{jo}, V_{ob}, and V_{ji}, so $C_v = 3$. As $\{01\}$ appears once in $E_{jo,ob}$ and $E_{ob,ji}$, $C_e = 2$. Hence, the query answer is $3-2=1$. In case that Q is an entry-based distinct-objects query, we need to consider two relevant short IDs, $\{01,10\}$. Each of these IDs appears once in V_{ob}, V_{ji} and V_{ci} and once in $E_{ob,ji}$ and $E_{ji,ci}$, so the number of entries for each of these IDs is $C_v - C_e = 3-2=1$. Hence, the query answer is two. EHSID also finds the exact query answers for both the entry-based cross-border and distinct-objects queries in this example.

3.7 Dummy Trajectories

Without relying on a trusted third party to perform anonymization, a mobile user can generate fake location trajectories, called *dummies*, to protect its privacy [33, 63]. Given a real user location trajectory T_r and a set of user-generated dummies T_d, the degree of privacy protection for the real trajectory is measured by the following metrics [63]:

1. **Snapshot disclosure (SD).** Let m be the number of location samples in T_r, S_i be the set of location samples in T_r and any T_d at time t_i, and $|S_i|$ be the size of S_i. SD is defined as the average probability of successfully inferring each true location sample in T_r, i.e., $SD = \frac{1}{m}\sum_{i=1}^{m} \frac{1}{|S_i|}$. Figure 15 gives a running example of $n = 3$

trajectories and $m = 5$ where T_r is from source location s_1 to destination location d_1 (i.e., $s_1 \rightarrow d_1$), T_{d_1} is $s_2 \rightarrow d_2$, and T_{d_2} is $s_3 \rightarrow d_3$. There are two intersections I_1 and I_2. At time t_1, since there are three different locations, i.e., $(1,2)$, $(1,4)$ and $(4,4)$, $|S_1| = 3$. At time t_2, T_r and T_{d_1} share one location, i.e., $(2,3)$, so $|S_2| = 2$. Thus, $SD = \frac{1}{5} \times (\frac{1}{3} + \frac{1}{2} + \frac{1}{2} + \frac{1}{3} + \frac{1}{3}) = \frac{2}{5}$.

2. **Trajectory disclosure (TD).** Given n trajectories, where k trajectories have intersection with at least one other trajectory and $n-k$ trajectories do not intersect any other trajectory, let N_k be the number of possible trajectories among the k trajectories. TD is defined as the probability of successfully identifying the true trajectory among all possible trajectories is $\frac{1}{N_k+(n-k)}$. In the running example (Figure 15), there are two intersection points I_1 and I_2, $k = 3$ and $N_k = 8$, i.e., there are eight possible trajectories: $s_1 \rightarrow I_1 \rightarrow d_2$, $s_1 \rightarrow I_1 \rightarrow I_2 \rightarrow d_1$, $s_1 \rightarrow I_1 \rightarrow I_2 \rightarrow d_3$, $s_2 \rightarrow I_1 \rightarrow d_2$, $s_2 \rightarrow I_1 \rightarrow I_2 \rightarrow d_1$, $s_2 \rightarrow I_1 \rightarrow I_2 \rightarrow d_3$, $s_3 \rightarrow I_2 \rightarrow d_1$, and $s_3 \rightarrow I_2 \rightarrow d_3$. Hence, $TD = \frac{1}{8+(3-3)} = \frac{1}{8}$.
3. **Distance deviation (DD).** DD is defined as the average distance between the i-th location samples of T_r and each T_{d_j}, i.e., $DD = \frac{1}{m}\sum_{i=1}^{m}(\frac{1}{n}\sum_{j=1}^{n} dist(T_r^i, T_{d_j}^i))$, where $dist(p,q)$ denotes the Euclidean distance between two point locations p and q. In the running example, $DD = \frac{1}{5} \times (2.80 + 0.71 + 0.71 + 2.12 + 2.12) = 1.69$.

Given a real trajectory T_r and the three user-specified parameters SD, TD, and DD in a privacy profile, the dummy-based anonymization algorithm incrementally uses DD to find a set of candidate dummies and select one with the best matching to SD and TD until it finds a set of trajectories (including T_r and selected dummies) that satisfies all the parameters [63]. Since a user can use consistent identities for its

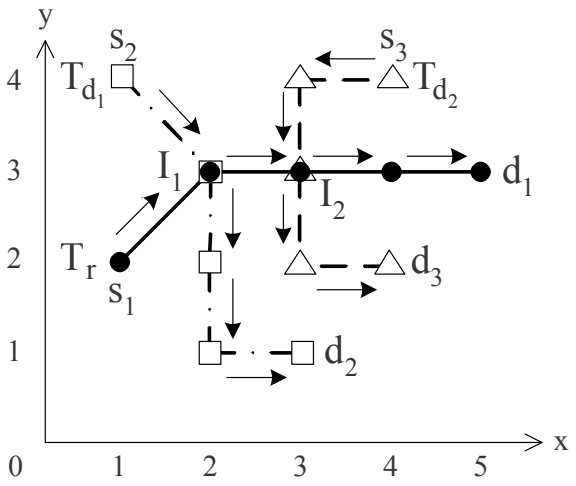


Fig. 15 One real trajectory T_r and two dummies T_{d_1} and T_{d_2}.

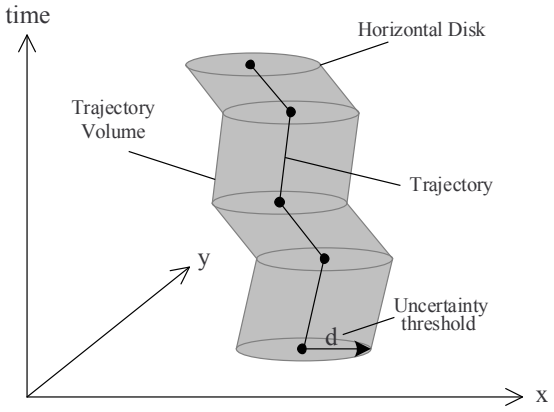


Fig. 16 The trajectory uncertainty model.

actual trajectory and other dummies, the dummy-based approach can support both Category-I and -II LBS, as depicted in Table 2.

4 Protecting Privacy in Trajectory Publication

In this section, we discuss privacy-preserving techniques for trajectory data publication. The anonymized trajectory data can be released to the public or third parties for answering spatio-temporal range queries and data mining. In the following sections, we present four trajectory anonymization techniques, namely, clustering-based, generalization-based, suppression-based and grid-based anonymization approaches, in Sections 4.1 to 4.4, respectively.

4.1 Clustering-based Anonymization Approach

The clustering-based approach [1] utilizes the uncertainty of trajectory data to group k co-localized trajectories within the same time period to form a k-anonymized aggregate trajectory. Given a trajectory T between times t_1 and t_n, i.e., $[t_1, t_n]$, and an uncertainty threshold d, each location sample in T, $p_i = (x_i, y_i, t_i)$, is modeled by a horizontal disk with radius d centered at (x_i, y_i). The union of all such disks constitutes the trajectory volume of T, as shown in Figure 16. Two trajectories T_p and T_q defined in $[t_1, t_n]$ are said to be co-localized with respect to d, if the Euclidean distance between each pair of points in T_p and T_q at time $t \in [t_1, t_n]$ is less than or equal to d. An anonymity set of k trajectories is defined as a set of at least k co-localized trajectories. The cluster of k co-localized trajectories is then transformed into an aggregate trajectory where each of its location points is computed by the

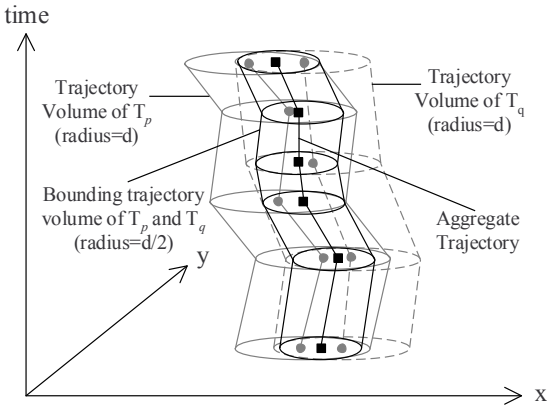


Fig. 17 Two-anonymized co-localized trajectories.

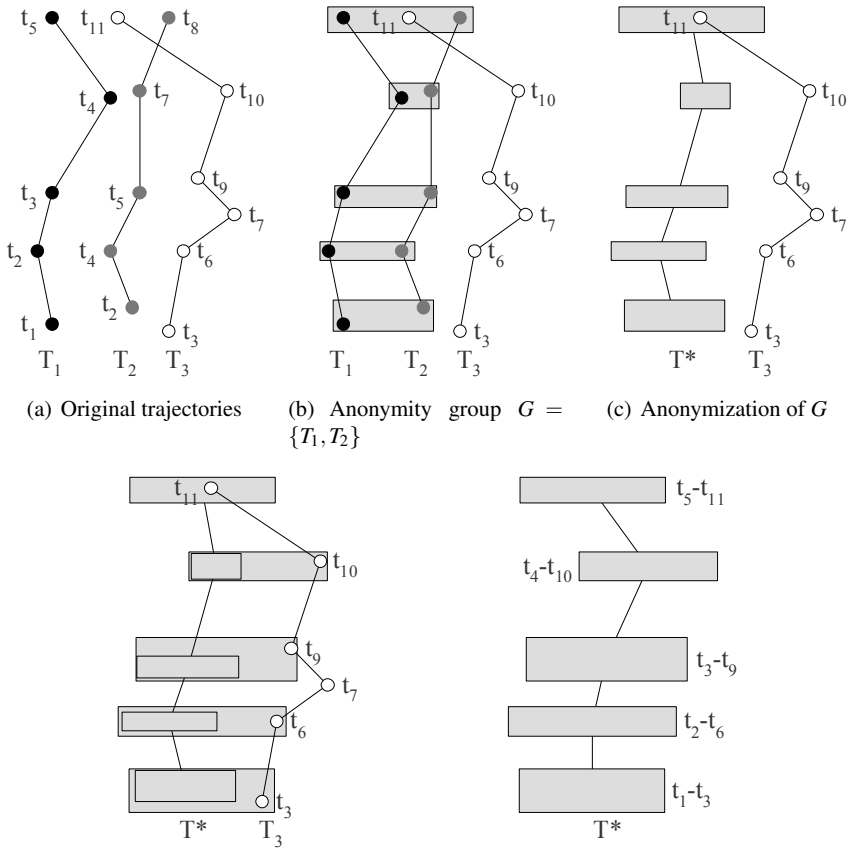
arithmetic mean of the location samples at the same time. Figure 17 gives the trajectory volumes of T_p and T_q that are represented by grey dotted lines, respectively. The trajectory volume with black lines is a bounding trajectory volume for T_p and T_q. The bounding trajectory volume is then transformed into an aggregate trajectory which is represented by the sequence of square markers.

The clustering-based anonymization algorithm consists of three main steps [1]:

1. **Pre-processing step.** The main task of this phase is to group all trajectories that have the same starting and ending times, i.e., they are in the same equivalence class with respect to time span. To increase the number of trajectories in an equivalence class, given an integer parameter π, all trajectories are trimmed if necessary such that only one timestamp every π can be the starting or ending point of a trajectory.
2. **Clustering step.** This phase clusters trajectories based on a greedy clustering scheme. For each equivalence class, a set of appropriate pivot trajectories are selected as cluster centers. For each cluster center, its nearest $k-1$ trajectories are assigned to the cluster, such that the radius of the bounding trajectory volume of the cluster is not larger than a certain threshold (e.g., $d/2$).
3. **Space transformation step.** Each cluster is transformed into a k-anonymized aggregate trajectory by moving all points at the same time to the corresponding arithmetic mean of the cluster.

4.2 Generalization-based Anonymization Approach

Since most data mining and statistical applications work on atomic trajectories, they are needed to be modified to work on aggregate trajectories generated by an anonymization algorithm (e.g., the clustering approach). To address this limitation,



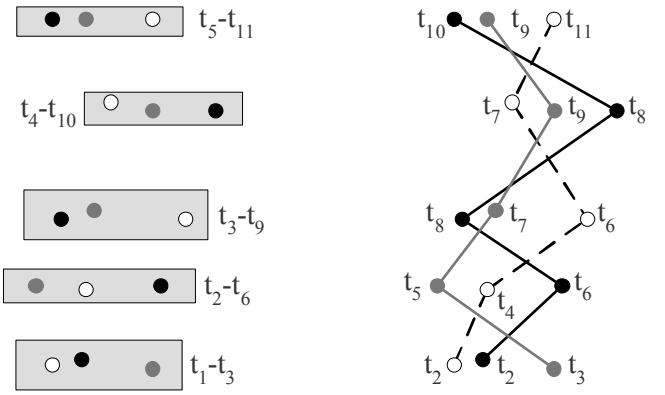
(a) Original trajectories (b) Anonymity group $G = \{T_1, T_2\}$ (c) Anonymization of G

(d) Add T_3 into G (e) Anonymization of G

Fig. 18 Generalization-based approach: Anonymization step.

the generalization-based algorithm first generalizes a trajectory data set into a set of k-anonymized trajectories, i.e., each one is a sequence of k-anonymized regions. Then, for each k-anonymized trajectory, the algorithm uniformly selects k atomic points from each anonymized region and links a unique atomic point from each anonymized region to reconstruct k trajectories [44]. More details about these two main steps are given below:

1. **Anonymization step.** Given a trajectory data set $\mathscr{T}$, each iteration of this step creates an empty anonymity group G and randomly samples one trajectory $T \in \mathscr{T}$. T is put into G as the group representative $Rep_G = T$. Then, the closest trajectory $T' \in \mathscr{T} - G$ to Rep_G is inserted into G and Rep_G is updated as the anonymization of Rep_G and T'. This anonymization process continues until G contains k trajectories. At the end of the iteration, the trajectories in G are



(a) Location samples selection (b) Trajectory reconstruction

Fig. 19 Generalization-based approach: Reconstruction step.

removed from $\mathscr{T}$. This step finishes when there are less than k remaining trajectories in $\mathscr{T}$. These trajectories are simply discarded.
Figure 18 gives an example of generalizing three trajectories T_1, T_2 and T_3 into a three-anonymized trajectory, where the timestamp of each location sample is shown beside its location. In this example, T_2 is first added into an empty group G as its representative Rep_G. Next T_1 is added to G and the location samples of T_1 and T_2 are generalized into a sequence of anonymized regions (represented by shaded rectangles), as depicted in Figure 18b. Rep_G is updated as the anonymization of T_1 and T_2, denoted by $T*$ (Figure 18c). T_3 is also added into G and a sequence of new anonymized regions is formed for G (Figure 18d). The time span of an anonymized region is the range from the smallest and largest timestamps of the location samples included in the region. Note that unmatched points (e.g., the location sample of T_3 at times t_7) are suppressed in this step. Since G already contains $k = 3$ trajectories, the anonymization process is done (Figure 18e).

2. **Reconstruction step.** Given a k-anonymized trajectory, k locations are uniformly selected in each of its anonymized regions, as illustrated in Figure 19a. Next, for each selected location, a timestamp is also uniformly selected from its associated time span. k trajectories are reconstructed by linking a unique location sample in each monitored region (Figure 19b).

The reconstructed trajectory data set can be released to the public or third parties for answering spatio-temporal queries and data analysis (e.g., data mining).

4.3 Suppression-based Anonymization Approach

Digital cash and electronic money have become very popular. People often use them to pay for their transportation and for their purchases at a wide variety of stores, e.g., convenient stores, grocery stores, restaurants and parking lots. Since a transaction is associated with the location of a particular store at a particular time, the sequence of a user's transactions can be considered as its trajectory information. Consider a digital cash or electronic money company publishes its original trajectory database T. A company A (e.g., 7-Eleven) that accepts its electronic payment service has part of trajectory information in T, i.e., the transactions are done by A. A can be considered as an adversary because A could use its knowledge T_A to identify its customers' private information [53].

Figure 20a shows an original trajectory database T that contains the location information of A's stores and other stores. A has part of the knowledge of T, T_A, i.e., the location information of its stores, as given in Figure 20b. By joining T_A to T, A knows that t_5^A actually corresponds to t_5 because t_5 is the only trajectory with $a_1 \rightarrow a_3$. Thus, A is 100% sure that the user of t_5^A visited b_1. Similarly, A knows that t_6^A corresponds to t_6, t_7 or t_8, so A can infer that the user of t_6^A visited b_2 with probability 66%, i.e., b_2 appears in t_7 and t_8.

The privacy protection for the above-mentioned linking attack is defined as: given an original trajectory database T and an adversary A's knowledge T_A, where T's locations take values from a data set P, T_A's locations take values from P_A and $P_A \subset P$, the probability that A can correctly identify the actual user of any location $p_i | p_i \in t_j \wedge p_i \notin t_k^A$ (where $t_j \in T$, $t_k^A \in T_A$, $p_i \in P$, and $p_i \notin P_A$) can be determined by the equation:

$$Pr(p_i, t_k^A, T) = \frac{|\{t' | t' \in S(t_k^A, T) \wedge p_i \in t' \wedge p_i \in P \wedge p_i \notin P_A\}|}{|S(t_k^A, T)|}, \tag{3}$$

where $S(t_k^A, T)$ denotes a set X of trajectories in T such that t_k^A is a subsequence of each trajectory in X. To protect user privacy, $Pr(p_i, t_k^A, T)$ should not be larger than a certain threshold δ. If $\delta = 50\%$, the publication of T makes privacy breaches, as given in the example (Figures 20a and 20b).

A greedy anonymization algorithm [53] is designed to iteratively suppress locations until the privacy constraint is met, i.e., $Pr(p_i, t_k^A, T) \leq \delta$ (computed by Equation 3) for each pair of $p_i \in P$ and $t_k^A \in T_A$. At the meantime, the utility of the published trajectory database is maximized. In the example, after the algorithm suppresses b_3 from t_2, a_1 from t_5, and b_3 from t_8, the transformed trajectory database T' ($\delta = 50\%$), as given in Figure 20c, can be published without compromising the user privacy.

One way to measure the utility is to compute the average difference between the original trajectories in T and the transformed ones in T' [53]. Let t be a trajectory in T and t' be its transformed form in T'. Let t_s and t_e (t'_s and t'_e) be the starting and ending locations of t (t'), respectively. The difference between t and t', denoted by

ID	Trajectory
t_1	$a_1 \rightarrow b_1 \rightarrow a_2$
t_2	$a_1 \rightarrow b_1 \rightarrow a_2 \rightarrow b_3$
t_3	$a_1 \rightarrow b_2 \rightarrow a_2$
t_4	$a_1 \rightarrow a_2 \rightarrow b_2$
t_5	$a_1 \rightarrow a_3 \rightarrow b_1$
t_6	$a_3 \rightarrow b_1$
t_7	$a_3 \rightarrow b_2$
t_8	$a_3 \rightarrow b_2 \rightarrow b_3$

(a) Original database T

ID	Trajectory
t_1^A	$a_1 \rightarrow a_2$
t_2^A	$a_1 \rightarrow a_2$
t_3^A	$a_1 \rightarrow a_2$
t_4^A	$a_1 \rightarrow a_2$
t_5^A	$a_1 \rightarrow a_3$
t_6^A	a_3
t_7^A	a_3
t_8^A	a_3

(b) Adversary's knowledge T_A

ID	Trajectory
t_1'	$a_1 \rightarrow b_1 \rightarrow a_2$
t_2'	$a_1 \rightarrow b_1 \rightarrow a_2$
t_3'	$a_1 \rightarrow b_2 \rightarrow a_2$
t_4'	$a_1 \rightarrow a_2 \rightarrow b_2$
t_5'	$a_3 \rightarrow b_1$
t_6'	$a_3 \rightarrow b_1$
t_7'	$a_3 \rightarrow b_2$
t_8'	$a_3 \rightarrow b_2$

(c) Transformed database T'

Fig. 20 Trajectory databases (Figure 1 in [53]).

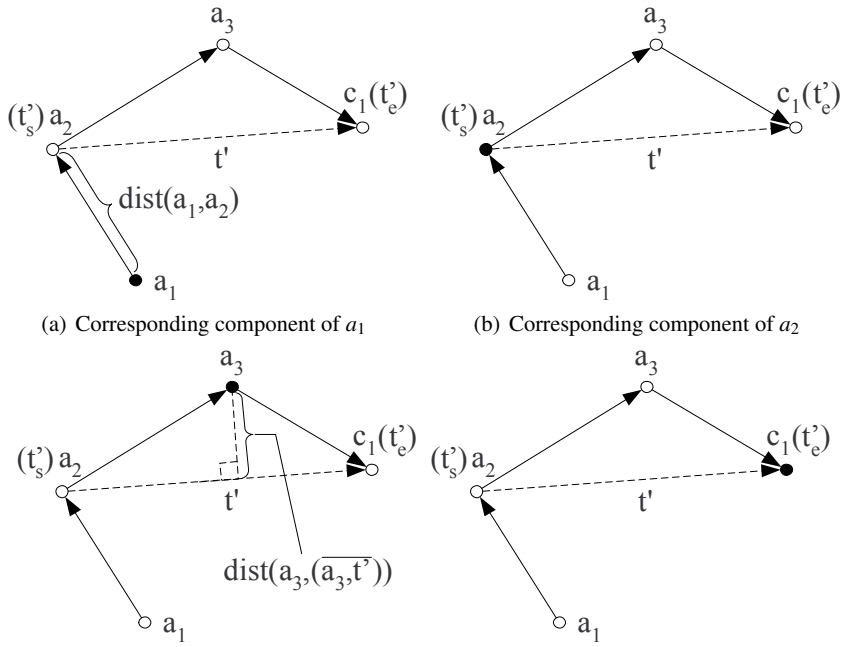
$diff(t,t')$, is computed as follows. For each location $p \in t$, it forms one component to the distance based on the following four cases:

1. If p is before t'_s, the corresponding component is $dist(p,t'_s)$.
2. If p is after t'_e, the corresponding component is $dist(p,t'_e)$.
3. If p is in-between t'_s and t'_e, the corresponding component is $dist(p,(\overline{p,t'}))$, where $(\overline{p,t'})$ is the perpendicular projection of p onto the transformed trajectory t'.
4. If $t' = \emptyset$, the corresponding component is set to the maximum distance between two locations on the map.

The final step of computing $diff(t,t')$ is to sum up the square of each component and take the root of the sum. Figure 21 gives a trajectory $t : a_1 \rightarrow a_2 \rightarrow a_3 \rightarrow c_1$ that is transformed to $t' : a_2 \rightarrow c_1$ by suppressing a_1 and a_3 from t. For the first location a_1 in t, Figure 21a shows that a_1 is before t'_s (i.e., Case 1), so its corresponding component is $d(a_1,t'_s) = d(a_1,a_2)$. Since a_2 is in-between t'_s and t'_e (i.e., Case 3), we need to find $(\overline{a_2,t'})$. As $(\overline{a_2,t'}) = a_2$, the corresponding component of a_2 is $dist(a_2,a_2) = 0$. a_3 is also in-between t'_s and t'_e (i.e., Case 3), so its corresponding component is $dist(a_3,(\overline{a_3,t'}))$. Similar to a_2, the corresponding component of c_1 is $dist(c_1,c_1) = 0$. Therefore, $diff(t,t') = \sqrt{[d(a_1,a_2)]^2 + [dist(a_3,(\overline{a_3,t'}))]^2}$.

4.4 Grid-based Anonymization Approach

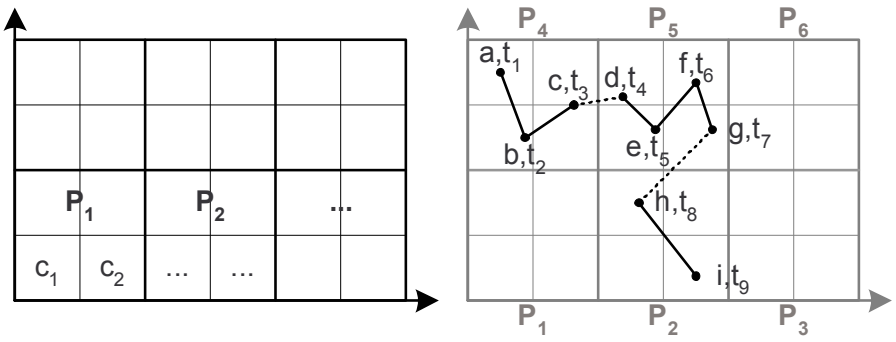
An grid-based anonymization approach is designed to anonymize user trajectories for privacy-preserving data mining [23]. This approach provides three anonymization features: spatial cloaking, temporal cloaking and trajectory splitting. Its basic idea is to construct a grid on the system space and partition the grid based on the required privacy constraint. Figure 22a shows part of the grid with 24 cells, i.e., $c_1, c_2, \ldots$. In this example, every four neighbor grid cells constitute a non-overlapping partition P. Generally speaking, we have to define a larger partition



(a) Corresponding component of a_1 (b) Corresponding component of a_2

(c) Corresponding component of a_3 (d) Corresponding component of c_1

Fig. 21 The difference between an original trajectory t and its transformed trajectory t'.



(a) Anonymization grid and partitioning (b) Trajectory anonymization

Fig. 22 Grid-based anonymization for trajectories.

area for users who need a higher level of privacy protection, but a smaller partition area for a lower level of privacy protection. However, larger partitions will lead to lower accuracy in the data mining results.

Figure 22b gives an example to illustrate how to anonymize a trajectory T : $\langle (a,t_1) \rightarrow (b,t_2) \rightarrow \ldots \rightarrow (i,t_9) \rangle$, where x and y in each pair (x,y) denote a location

and its update timestamp, respectively, to a list of anonymization rectangles. Locations a, b, and c are in the same grid cell P_4, so they are blurred into the anonymization rectangle P_4 with a time range from t_1 to t_3, i.e., $(P_4, [t_1, t_3])$. Since t crosses two partitions, from P_4 to P_5, the sub-path from c to d is ignored. T is split into two sub-trajectories. Similarly, locations d, e, f, and g are transformed into $(P_5, [t_4, t_7])$. As the path gh crosses two partitions, it is discarded and T is split again. Finally, locations h and i are transformed into $(P_2, [t_8, t_9])$. Therefore, the transformed form of t is $\langle (P_4, [t_1, t_3]) \rightarrow (P_5, [t_4, t_7]) \rightarrow (P_2, [t_8, t_9]) \rangle$. Trajectories anonymized by the grid-based approach can answer spatio-temporal density queries [23].

5 Conclusion

Location privacy protection in continuous location-based services (LBS) and trajectory data publication has drawn a lot of attention from the industry and academia. It is expected that more effective and efficient privacy preserving technologies will be developed in the near future. We want to provide some future directions in these two problems as the conclusion of this chapter. For continuous LBS, new privacy-preserving techniques are needed to protect personalized LBS. This is because personalized LBS require more user semantics, e.g., user preferences and background information, such as salary and occupation, rather than just some simple query parameters, such as a distance range and an object type of interest. An adversary could use such user semantics to infer the user location with higher confidence. For example, suppose an adversary knows that a target user Alice usually has dinner from 6pm to 7pm during weekdays and she does not like Japanese and Thailand food. Given a cloaked spatial region of Alice's location at 6:30pm on Monday and the region contains two Japanese restaurants, one Thailand restaurant and one Chinese restaurant, the adversary can infer that Alice in the Chinese restaurant with very high confidence. Existing privacy-preserving techniques for location trajectory publication only support simple aggregate analysis, such as range queries and clustering. Researchers should develop new trajectory anonymization techniques that support more useful and complex spatio-temporal queries (e.g., how many vehicles travel from a shopping mall to a cinema from 1pm to 2pm during weekends, their most popular paths, and their average travel time) and spatio-temporal data analysis.

References

1. Abul, O., Bonchi, F., Nanni, M.: Never walk alone: Uncertainty for anonymity in moving objects databases. In: Proceedings of the IEEE International Conference on Data Engineering (2008)
2. Bamba, B., Liu, L., Pesti, P., Wang, T.: Supporting anonymous location queries in mobile environments with PrivacyGrid. In: Proceedings of the International Conference on World Wide Web (2008)

3. Bao, J., Chow, C.Y., Mokbel, M.F., Ku, W.S.: Efficient evaluation of k-range nearest neighbor queries in road networks. In: Proceedings of the International Conference on Mobile Data Management (2010)
4. Beresford, A.R., Stajano, F.: Location privacy in pervasive computing. IEEE Pervasive Computing **2**(1), 46–55 (2003)
5. Cheng, R., Zhang, Y., Bertino, E., Prabhakar, S.: Preserving user location privacy in mobile data management infrastructures. In: Proceedings of International Privacy Enhancing Technologies Symposium (2006)
6. Chow, C.Y., Bao, J., Mokbel, M.F.: Towards location-based social networking services. In: Proceedings of the ACM SIGSPATIAL International Workshop on Location Based Social Networks (2010)
7. Chow, C.Y., Mokbel, M., He, T.: A privacy-preserving location monitoring system for wireless sensor networks. IEEE Transactions on Mobile Computing **10**(1), 94–107 (2011)
8. Chow, C.Y., Mokbel, M.F.: Enabling private continuous queries for revealed user locations. In: Proceedings of the International Symposium on Spatial and Temporal Databases (2007)
9. Chow, C.Y., Mokbel, M.F., Aref, W.G.: Casper*: Query processing for location services without compromising privacy. ACM Transactions on Database Systems **34**(4), 24:1–24:48 (2009)
10. Chow, C.Y., Mokbel, M.F., Bao, J., Liu, X.: Query-aware location anonymization in road networks. GeoInformatica **15**(3), 571–607 (2011)
11. Chow, C.Y., Mokbel, M.F., Liu, X.: A peer-to-peer spatial cloaking algorithm for anonymous location-based services. In: Proceedings of the ACM Symposium on Advances in Geographic Information Systems (2006)
12. Chow, C.Y., Mokbel, M.F., Liu, X.: Spatial cloaking for anonymous location-based services in mobile peer-to-peer environments. GeoInformatica **15**(2), 351–380 (2011)
13. Dateline NBC: Tracing a stalker. `http://www.msnbc.msn.com/id/19253352` (2007)
14. Duckham, M., Kulik, L.: A formal model of obfuscation and negotiation for location privacy. In: Proceedings of International Conference on Pervasive Computing (2005)
15. FoxNews: Man accused of stalking ex-girlfriend with GPS. `http://www.foxnews.com/story/0,2933,131487,00.html` (2004)
16. Freudiger, J., Raya, M., Felegyhazi, M., Papadimitratos, P., Hubaux, J.P.: Mix-zones for location privacy in vehicular networks. In: Proceedings of the International Workshop on Wireless Networking for Intelligent Transportation Systems (2007)
17. Freudiger, J., Shokri, R., Hubaux, J.P.: On the optimal placement of mix zones. In: Proceedings of International Privacy Enhancing Technologies Symposium (2009)
18. Fung, B.C.M., Wang, K., Chen, R., Yu, P.S.: Privacy-preserving data publishing: A survey of recent developments. ACM Computing Surveys **42**(4), 14:1–14:53 (2010)
19. Gedik, B., Liu, L.: Protecting location privacy with personalized k-anonymity: Architecture and algorithms. IEEE Transactions on Mobile Computing **7**(1), 1–18 (2008)
20. Ghinita, G., Kalnis, P., Khoshgozaran, A., Shahabi, C., Tan, K.L.: Private queries in location based services: Anonymizers are not necessary. In: Proceedings of the ACM Conference on Management of Data (2008)
21. Ghinita, G., Kalnis, P., Skiadopoulos, S.: PRIVÉ: Anonymous location-based queries in distributed mobile systems. In: Proceedings of the International Conference on World Wide Web (2007)
22. Ghinita1, G., Kalnis, P., Skiadopoulos, S.: MobiHide: A mobile peer-to-peer system for anonymous location-based queries. In: Proceedings of the International Symposium on Spatial and Temporal Databases (2007)
23. Gidófalvi, G., Huang, X., Pedersen, T.B.: Privacy-preserving data mining on moving object trajectories. In: Proceedings of the International Conference on Mobile Data Management (2007)
24. Google Geocoding API: `http://code.google.com/apis/maps/documentation/geocoding/`

25. Gruteser, M., Grunwald, D.: Anonymous usage of location-based services through spatial and temporal cloaking. In: Proceedings of the International Conference on Mobile Systems, Applications, and Services (2003)
26. Gruteser, M., Hoh, B.: On the anonymity of periodic location samples. In: Proceedings of the International Conference on Security in Pervasive Computing (2005)
27. Hoh, B., Gruteser, M., Xiong, H., Alrabady, A.: Achieving guaranteed anonymity in GPS traces via uncertainty-aware path cloaking. IEEE Transactions on Mobile Computing **9**(8), 1089–1107 (2010)
28. Hong, J.I., Landay, J.A.: An architecture for privacy-sensitive ubiquitous computing. In: Proceedings of the International Conference on Mobile Systems, Applications, and Services (2004)
29. Hu, H., Lee, D.L.: Range nearest-neighbor query. IEEE Transactions on Knowledge and Data Engineering **18**(1), 78–91 (2006)
30. Ilarri, S., Mena, E., Illarramendi, A.: Location-dependent query processing: Where we are and where we are heading. ACM Computing Surveys **42**(3), 12:1–12:73 (2010)
31. Kalnis, P., Ghinita, G., Mouratidis, K., Papadias, D.: Preventing location-based identity inference in anonymous spatial queries. IEEE Transactions on Knowledge and Data Engineering **19**(12), 1719–1733 (2007)
32. Khoshgozaran, A., Shahabi, C.: Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy. In: Proceedings of the International Symposium on Spatial and Temporal Databases (2007)
33. Kido, H., Yanagisawa, Y., Satoh, T.: An anonymous communication technique using dummies for location-based services. In: Proceedings of IEEE International Conference on Pervasive Services (2005)
34. Ku, W.S., Zimmermann, R., Peng, W.C., Shroff, S.: Privacy protected query processing on spatial networks. In: Proceedings of the International Workshop on Privacy Data Management (2007)
35. LeFevre, K., DeWitt, D., Ramakrishnan, R.: Mondrian multidimensional *k*-anonymity. In: Proceedings of the IEEE International Conference on Data Engineering (2006)
36. Li, N., Li, T., Venkatasubramanian, S.: Closeness: A new privacy measure for data publishing. IEEE Transactions on Knowledge and Data Engineering **22**(7), 943–956 (2010)
37. Ma, C.Y., Yau, D.K.Y., Yip, N.K., Rao, N.S.V.: Privacy vulnerability of published anonymous mobility traces. In: Proceedings of the ACM International Conference on Mobile Computing and Networking (2010)
38. Machanavajjhala, A., Kifer, D., Gehrke, J., Venkitasubramaniam, M.: *l*-diversity: Privacy beyond *k*-anonymity. ACM Transactions on Knowledge Discovery from Data **1**(1), 3:1–3:52 (2007)
39. Marist Institute for Public Opinion (MIPO): Half of Social Networkers Online Concerned about Privacy. `http://maristpoll.marist.edu/714-half-of-social-networkers-online-%concerned-about-privacy/`. July 14, 2010
40. Meyerowitz, J., Choudhury, R.R.: Hiding stars with fireworks: Location privacy through camouflage. In: Proceedings of the ACM International Conference on Mobile Computing and Networking (2009)
41. Mokbel, M.F., Chow, C.Y., Aref, W.G.: The new casper: Query procesing for location services without compromising privacy. In: Proceedings of the International Conference on Very Large Data Bases (2006)
42. Mokbel, M.F., Levandoski, J.: Towards context and preference-aware location-based database systems. In: Proceedings of the ACM International Workshop on Data Engineering for Wireless and Mobile Access (2009)
43. Mouratidis, K., Yiu, M.L.: Anonymous query processing in road networks. IEEE Transactions on Knowledge and Data Engineering **22**(1), 2–15 (2010)
44. Nergiz, M.E., Atzori, M., Saygin, Y., Güç, B.: Towards trajectory anonymization: A generalization-based approach. Transactions on Data Privacy **2**(1), 47–75 (2009)

45. Palanisamy, B., Liu, L.: Mobimix: Protecting location privacy with mix zones over road networks. In: Proceedings of the IEEE International Conference on Data Engineering (2011)
46. Pan, X., Meng, X., Xu, J.: Distortion-based anonymity for continuous queries in location-based mobile services. In: Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (2009)
47. Pfitzmann, A., Kohntopp, M.: Anonymity, unobservability, and pseudonymity - a proposal for terminology. In: Proceedings of the Workshop on Design Issues in Anonymity and Unobservability (2000)
48. Reid, D.: An algorithm for tracking multiple targets. IEEE Transactions on Automatic Control **24**(6), 843–854 (1979)
49. Samarati, P.: Protecting respondents identities in microdata release. IEEE Transactions on Knowledge and Data Engineering **13**(6), 1010–1027 (2001)
50. Sun, C., Agrawal, D., Abbadi, A.E.: Exploring spatial datasets with histograms. In: Proceedings of the IEEE International Conference on Data Engineering (2002)
51. Sweeney, L.: Achieving *k*-anonymity privacy protection using generalization and suppression. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems **10**(5), 571–588 (2002)
52. Sweeney, L.: *k*-anonymity: A model for protecting privacy. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems **10**(5), 557–570 (2002)
53. Terrovitis, M., Mamoulis, N.: Privacy preservation in the publication of trajectories. In: Proceedings of the International Conference on Mobile Data Management (2008)
54. USAToday: Authorities: GPS system used to stalk woman. `http://www.usatoday.com/tech/news/2002-12-30-gps-stalker_x.htm` (2002)
55. Voelcker, J.: Stalked by satellite: An alarming rise in gps-enabled harassment. IEEE Spectrum **47**(7), 15–16 (2006)
56. Wang, T., Liu, L.: Privacy-aware mobile services over road networks. In: Proceedings of the International Conference on Very Large Data Bases (2009)
57. Webroot Software, Inc.: Webroot survey finds geolocation apps prevalent amongst mobile device users, but 55% concerned about loss of privacy. `http://pr.webroot.com/threat-research/cons/social-networks-mobile-security-071310.html`. July 13, 2010
58. Xiao, X., Yi, K., Tao, Y.: The hardness and approximation algorithms for l-diversity. In: Proceedings of the International Conference on Extending Database Technology (2010)
59. Xie, H., Kulik, L., Tanin, E.: Privacy-aware traffic monitoring. IEEE Transactions on Intelligent Transportation Systems **11**(1), 61–70 (2010)
60. Xu, T., Cai, Y.: Location anonymity in continuous location-based services. In: Proceedings of the ACM Symposium on Advances in Geographic Information Systems (2007)
61. Xu, T., Cai, Y.: Exploring historical location data for anonymity preservation in location-based services. In: Proceedings of IEEE INFOCOM (2008)
62. Yiu, M.L., Jensen, C., Huang, X., Lu, H.: Spacetwist: Managing the trade-offs among location privacy, query performance, and query accuracy in mobile services. In: Proceedings of the IEEE International Conference on Data Engineering (2008)
63. You, T.H., Peng, W.C., Lee, W.C.: Protecting moving trajectories with dummies. In: Proceedings of the International Workshop on Privacy-Aware Location-Based Mobile Services (2007)
64. Zhang, C., Huang, Y.: Cloaking locations for anonymous location based services: A hybrid approach. GeoInformatica **13**(2), 159–182 (2009)