Differentially Private Publication of General Time-Serial Trajectory Data

Jingyu Hua, Yue Gao and Sheng Zhong
State Key Laboratory for Novel Software Technology,
Department of Computer Science and Technology, Nanjing University, China
Email: huajingyu@nju.edu.cn, njucsmoon@smail.nju.edu.cn, zhongsheng@nju.edu.cn

***Abstract*—Trajectory data, i.e., human mobility traces, is extremely valuable for a wide range of mobile applications. However, publishing raw trajectories without special sanitization poses serious threats to individual privacy. Recently, researchers begin to leverage differential privacy to solve this challenge. Nevertheless, existing mechanisms make an implicit assumption that the trajectories contain a lot of identical prefixes or n-grams, which is not true in many applications. This paper aims to remove this assumption and propose a differentially private publishing mechanism for more general time-series trajectories. One natural solution is to generalize the trajectories, i.e., merge the locations at the same time. However, trivial merging schemes may breach differential privacy. We, thus, propose the first differentially-private generalization algorithm for trajectories, which leverage a carefully-designed exponential mechanism to probabilistically merge nodes based on trajectory distances. Afterwards, we propose another efficient algorithm to release trajectories after generalization in a differential private manner. Our experiments with real-life trajectory data show that the proposed mechanism maintains high data utility and is scalable to large trajectory datasets.**

***Index Terms*—Trajectory, Differential Privacy, Data Publishing**

I. Introduction

Trajectory data, i.e., human mobility traces, is valuable for a variety of areas [2], [23]. For example, in urban planning, it can help build better city transportation system and prevent frequent traffic jams. In online social networking, it can help better understand how people develop social relations, and create more worthy applications. Thanks to the popularization of mobile devices (i.e., smartphones and PDAs) supporting GPS as well as other positioning facilities, nowadays, it is really easy for service providers and mobile phone producers to collect mobility traces of users.

While trajectory data provides great benefits, it poses serious threats to individual privacy [6], [16]. After aggregating sufficient data, the collector will usually publish it to internal departments or external partners for various analysis purposes [4]. Even if we can guarantee the security and the legality of the collection process, we cannot exclude the possibility that there exist malicious guys who are curious about user privacy in these departments and partners. If trajectory data is published in an inappropriate form, adversaries may leverage background knowledge to link mobility traces to individuals, and further infer sensitive personal information like health condition, religious and sexual preferences. In this sense, we call for a privacy preserving publishing mechanism for trajectory data.

Strictly speaking, discrete trajectory data (i.e., spatio-temporal data) can be considered as relational data. Therefore, most of early solutions [1], [19], [22], [9], [15], [5] rely on partition-based privacy models [8] (e.g., k-anonymity [18] and confidence bounding [19], [5]), which are widely used in anonymization of relational data. Unfortunately, these schemes have been found to be vulnerable to many types of privacy attacks, such as composition attack [8], deFinetti attack [11] and foreground knowledge attack [21], and are unsuitable for publishing trajectories [14].

To avoid these problem, *differential privacy* [7] is recently introduced to privacy preserving data publishing. This novel privacy model make no assumptions on adversaries background knowledge, and requires that any computation on the underlying database is insensitive to the change (insertion or deleting) of a single record. Thus, if the publication process satisfies differential privacy, it can guarantee that the released data will not breach an individual's privacy regardless of whether her record is present or absent in the original data. Chen et al. [4] propose the first differentially private trajectory-publishing mechanism. They elegantly employ a noisy prefix tree based on the underlying data to narrow down the output domain, which guarantees the high efficiency of the publication process. They also make use of two sets of inherent constraints of a prefix tree to conduct constrained inferences, which helps improve the utility of the release. They have extended this approach by using the variable-length n-gram model to sanitize general sequential data [3].

Nevertheless, these two proposals have to *make an implicit assumption that the raw trajectories to be published contain a large number of common prefixes or n-grams (i.e., sub-trajectories of length n)*. This is because the anonymization will add noises to the real counts of prefixes or n-grams to guarantee differential privacy. If these counts are very small, the added noises become relatively large and can publish a lot of meaningless data. Unfortunately, in the real world, *it does exist a lot of trajectory datasets do not follow this assumption*. For instance, in many cases, trajectories are sequences of fine-grained geographic coordinates obtained from positioning systems such as GPS. As the domain of such coordinates are

Sheng Zhong is the corresponding author. Sheng Zhong was supported in part by RPGE, NSFC-61321491, NSFC-61425024, and NSFC-61300235. Part of the work was done while supported in part by NSF-0845149.

extremely huge, it may be difficult to find any two trajectories sharing the same prefix or n-gram. In addition, trajectory data in many applications attaches a timestamp to each location. Chen et al. claim that their proposals can easily overcome this challenge by labeling each node in a trajectory by both a location and a timestamp. However, this measure may further reduce the real counts of prefixes (or n-grams) as two prefixes (or n-grams) with the same sequence of locations but different timestamps are considered different.

Contributions. In this paper, we aim to remove the assumption and propose a practical non-intensive differentially private publishing mechanism for general time-series trajectory data. A straightforward solution is to generalize the locations by merging close ones at the same timestamps. Nevertheless, ordinary merging mechanisms may breach the differential privacy. We solve this challenge and make the following contributions:

(1) We propose the first differentially private location generalization algorithm for removing the dependency of existing mechanisms on the implicit assumption. This algorithm leverages an exponential mechanism to probabilistically merge locations at the same time points. We propose an optimization based on clustering to avoid exhaustively accounting every merging option, which significantly improves the time efficiency. In addition, as this mechanism prefers to merging locations belonging to closer trajectories, it prevents from bringing heavy side effects on the data utility during generalization.

(2) We then propose a simple and efficient differentially private algorithm to publish trajectories after generalization. This algorithm needs neither to explicitly consider each possible trajectory over the generalized location domains, nor to construct any prefix tree. As a result, its time efficiency is extremely high. It also introduce a distance constraint between two continuous locations to guarantee the data utility. We formally prove that this scheme satisfies ϵ-differential privacy.

(3) We conduct experiments to evaluate the proposed mechanism with a real-life dataset containing more than 6000 taxi trajectories (each is composed of locations of 32 timestamps). We examine the utility of the published dataset by measuring its *Hausdorff Distance* to the original dataset, and performing spatio-temporal range queries. We also measure the time efficiency of the proposed mechanism. The results demonstrate that our mechanism maintains good utility and is scalable to large volume of real-life trajectory data.

II. Related Work

According to [17], [10], existing trajectory publishing mechanisms can be classified into two types. The first type aims to sanitize and publish a trajectory dataset containing a great number of trajectories. Each trajectory is regarded as one record. The second type [17], [10], however, aims to publish only one trajectory. Each position in this trajectory is considered as one record. The major difference between these two types is that the first one focuses on protecting the trajectory privacy while the second one focuses on protecting the position privacy in a trajectory. In this paper, we focus on the trajectory privacy that our proposal belongs to the first type.

Recent mechanisms of the first type can be further divided into two subcategories based on what privacy models they use. The first subcategory [1], [19], [22], [9], [15], [5] makes use of the *partition-based privacy models* [8] such as k-*anonymity* [18] as well as its variants. They usually first cluster records in the database into disjoint groups satisfying some privacy constraints, and then calculate and publish certain statistics for each group. For instance, Abul et al. [1] propose the (k, δ)-anonymity model, which is generalized from k-anonymity. Its basic idea is to modify the original trajectories based on clustering and space translation so that at least k different trajectories co-exist in a cylinder with radius δ. It then publishes the cylinders instead of the trajectories to protect the privacy of moving objects. Unfortunately, a lot of work [8], [11], [21] has demonstrated that the partition-based privacy models are vulnerable to a lot of attacks. As a result, they are considered to be unable to provide sufficient privacy protection in trajectory publication [4].

Due to the drawbacks of the partition-based privacy models, differential privacy, which is one kind of *randomization-based privacy model* [8], is recently employed for non-interactive privacy preserving data publishing [14]. This model makes no assumption about an adversary's background knowledge and can guarantee that the output is insensitive to any individual's data. Chen et al. [4] first apply this model to trajectory publishing. Previous differentially private data publishing approaches are all data independent, which means they have to consider all the entries in the output domain regardless of the underlying database. This is computationally infeasible for high-dimensional data such as trajectories since the output domain is extremely huge. Chen et al. thereby propose an efficient data-dependent mechanism. To guarantee the efficiency, they narrow down the output domain by recursively building a noisy prefix tree based on the underlying data. Specifically, all the trajectories with the identical prefix are grouped into the same branch. One branch is pruned if the noisy count of its trajectories is below a predefined threshold.

Unfortunately, with the growth of the prefix tree, the number of sequences falling into a branch decreases quickly, resulting in poor utility [3]. Chen et al. [3] then turns to make use of the variable n-gram model as well as a set of novel techniques based on the Markov assumption to improve the utility. Nevertheless, both these two proposals have to make an implicit assumption that the raw trajectories to be published contain a large number of common prefixes or n-grams, which is not true in many applications.

III. Preliminaries

In this section, we define the time-series trajectory database, review the theory of differential privacy and finally present a problem statement.

A. Time-Series Trajectory and Database

In this paper, a time-series trajectory is defined below:

Definition 1 (Time-Series Trajectory)**.** *A trajectory is an ordered list of time-location pairs:* $T = (t_1, l_1) \rightarrow (t_2, l_2) \rightarrow \cdots \rightarrow (t_{|T|}, l_{|T|})$*, where* $|T|$ *is the length of this trajectory and* $\forall i (1 \leq i \leq |T|)$*,* $l_i \in \Gamma_i$ *is a discrete spatial point, which is represented by the latitude and longitude coordinate.* Γ_i *is the universe of locations at time* T_i*.*

Each trajectory represents the movement history of a human being. We denote by $Time(T)$ the set of timestamps in T, namely, $Time(T) = \{t_1, t_2, \cdots, t_{|T|}\}$, and by $T(t_i)$ the location at time t_i in T, namely $T(t_i) = l_i$.

A trajectory database $\mathbb{D}$ of size $|\mathbb{D}|$ is a multiset of time-series trajectories $\mathbb{D} = \{T_1, T_2, \cdots, T_{|\mathbb{D}|}\}$. For simplicity, we assume that the trajectories in $\mathbb{D}$ are recorded for the same set of time points, i.e., $\forall 1 \leq i, j \leq |\mathbb{D}|$, $Time(T_i) = Time(T_j)$. In $\mathbb{D}$, the location universe at time T_i, Γ_i, is defined to be the set $\{T_k.l_i | k = 1, 2, \cdots, |T|\}$.

Since every location is a discrete spatial point, we think it is difficult to find two trajectories $T_i, T_j \in \mathbb{D}$ that share the same location at any time point, i.e., $\forall t \in Time(T_i)$, $T_i(t) \neq T_j(t)$.

B. Differential Privacy

Differential privacy becomes a hot privacy model recently as it provides a strong privacy guarantee. It requires that the output of any computation based on a underlying database is insensitive to the removal and the addition of a particular record. This indicates that it is privacy harmless for a record owner to include his record in the database. We formally give the definition as well as some important properties of differential privacy below [14].

Definition 2 (ϵ-differential privacy)**.** *A randomized algorithm* Ag *is differentially private if and only if any two databases* $\mathbb{D}$ *and* $\mathbb{D}'$ *contain at most one different record (i.e.,* $|\mathbb{D} \triangle \mathbb{D}'| \leq 1$*), and for any possible anonymized output* $O \in Range(Ag)$*,*

$$Pr[Ag(\mathbb{D}) = O] \leq e^{\epsilon} \times Pr[Ag(\mathbb{D}') = O]$$

where the probability is taken over the randomness of Ag.

In the above equation, ϵ is positive, and it is believed that the smaller its value is, the stronger the privacy guarantee is. Generally, its value is as small as 0.1 or even smaller.

To reach differential privacy, the standard approach is to add random noise to the true output of the function upon the underling database. Its basic idea is to use the added noise to hide the output difference due to the single record change in the underling database. Therefore, the noise values are determined by the *sensitivity* of the output function [14], i.e., the maximal difference of outputs from databases containing at most one different record.

Definition 3 (Global Sensitivity)**.** *For a given function* $f : \mathbb{D} \rightarrow R^d$ *, its sensitivity is*

$$\triangle f = \max_{\mathbb{D}, \mathbb{D}'} \|f(\mathbb{D}) - f(\mathbb{D}')\|_1,$$

for all $\mathbb{D}$ *and* $\mathbb{D}'$ *differing in one record.*

Based on this fundamental concept, two major techniques for achieving differential privacy have been proposed, namely Laplace mechanism [7] and exponential mechanism [13].

Laplace Mechanism. This mechanism is designed for the functions whose outputs are real. It adds proper Laplace noises to the real outputs to achieve differential privacy. Specifically, the noise is generated according to a Laplace distribution Lap(λ) with the probability dense function $Pr(x|y) = \frac{1}{2\lambda} e^{\frac{-|x|}{\lambda}}$, where the parameter λ is determined by the global sensitivity $\triangle f$ and the desired differential privacy parameter ϵ. The following theorem presents the concrete relation between these parameters.

Theorem 1. *[7] For any* $f : D \rightarrow R^d$*, the mechanism that adds independently generated noises with distribution* $Lap(\lambda)$ *to each of the* d *outputs satisfies* ϵ*-differential privacy if* $\epsilon = \triangle f / \epsilon$*.*

Exponential Mechanism. There are many functions whose outputs are not real or make no sense after being added noises. For instance, some record attributes may be non-numeric. McSherry and Talwar [13] propose the exponential mechanism to achieve differential privacy in this case. It first defines a utility function $u : (\mathbb{D} \times \tau) \rightarrow R$ that assigns a real valued score to each output r in the output domain R. Here, a higher score indicates a better utility. It then selects an output $r \in R$ with the probability proportional to $e^{\frac{\epsilon u(\mathbb{D}, r)}{2 \triangle u}}$, where $\triangle u = \max_{\forall r, \mathbb{D}, \mathbb{D}'} |u(\mathbb{D}, r) - u(\mathbb{D}', r)|$ is the sensitivity of the utility function. As the outputs with higher scores are more likely to be selected, this mechanism is close to the optimal with respect to u. In addition, the utility function should be insensitive to the changes of a single record.

Theorem 2. *[13] For any function* $u : (D \times \tau) \rightarrow R$ *, the mechanism chooses an output* $r \in R$ *with the probability proportional to* $e^{\frac{\epsilon u(\mathbb{D}, r)}{2 \triangle u}}$ *can guarantee* ϵ*-differential privacy.*

Composition Properties. Differential privacy has two important composition properties [12]. First, a sequence of computations that each provides differential privacy independently also provides differential privacy as a whole but the privacy cost (represented by ϵ in Definition 2) is accumulated. This is known as *sequential composition*. Second, if the sequence of computations are performed on disjoint sub-databases, the final privacy cost is not accumulated but determined by the computation provides the worst privacy guarantee. This is known as *parallel composition*. We use the following two theorems [12] to formally describe these two properties.

Theorem 3 (Sequential composition)**.** *Suppose that each algorithm* Ag_i *provides* ϵ_i*-differential privacy. A sequence of* Ag_i *over a database* $\mathbb{D}$ *provides* $\sum \epsilon_i$*-differential privacy as a whole.*

Theorem 4 (Parallel composition)**.** *Suppose that each algorithm* Ag_i *provides* ϵ*-differential privacy. A sequence of* Ag_i *over a set of disjoint data sets* $\mathbb{D}_i$ *provides* ϵ*-differential privacy as a whole.*

C. Problem Statement

Suppose a data owner wants to publish a trajectory database $\mathbb{D}$ for analysis. Our objective is to anonymize $\mathbb{D}$ such that the generated database $\tilde{\mathbb{D}}$ meets the following two requirements.

(1) **Guaranteeing ϵ-differential privacy**: Formally, for any trajectory database $\mathbb{D}'$ differing in one trajectory with $\mathbb{D}$, the anonymization mechanism A satisfies $e^{-\epsilon} \times Pr[A(D) = \tilde{D}] \leq Pr[A(D) = \tilde{D}] \leq e^{\epsilon} \times Pr[A(D) = \tilde{D}]$.

(2) **Guaranteeing a high data utility**: Differently private mechanisms usually have to add noise to the original data. As a result, the data utility is inevitably affected. The proposed anonymization mechanism should impose as fewer negative impacts on the data utility as possible. We formally describe the data utility below.

We require that the proposed mechanism release as many trajectories as the original dataset $\mathbb{D}$, i.e., $|\tilde{\mathbb{D}}| = |\mathbb{D}|$. Then, we leverage *Hausdorff Distance*, which is widely used in machine learning tasks to measure the similarity between two non-empty sets of points, to measure the utility of $\tilde{\mathbb{D}}$ with respect to $\mathbb{D}$:

$$Utility(\tilde{\mathbb{D}}) = \max\{h(\tilde{\mathbb{D}}, \mathbb{D}), h(\mathbb{D}, \tilde{\mathbb{D}})\}, \tag{1}$$

where $h(\tilde{\mathbb{D}}, \mathbb{D}) = \max_{T \in \tilde{\mathbb{D}}} \big\{ \min_{T' \in \mathbb{D}} \{Distance(T, T')\} \big\}$. Obviously, a smaller $Utility(\tilde{\mathbb{D}})$ indicates a higher data utility.

IV. Our Proposal

In this section, we present the design of the proposed differentially private publishing mechanism for general time-series trajectories. Specially, we first present the overview of our proposal. We then elaborate the two key algorithms in details, and prove that the proposed mechanism satisfies ϵ-differential privacy.

A. Overview

A naive anonymization scheme that can achieve differential privacy is to output the noisy count of every possible entry in the trajectory universe regardless of the underlying database. However, as Chen et al. [4] pointed out, this scheme is computationally infeasible due to the extremely high dimension and the large location universe of trajectory data. Chen et al. [4], [3] address this challenge by constructing a prefix tree or a exploration tree based on the underlying database to narrow down the output domain. However, as we mentioned earlier, their approaches have to assume that the trajectories share many identical prefixes or n-grams, which is no longer true in many applications. We consider more general trajectories whose locations are fine grained. The frequencies of individual trajectories, prefixes as well as n-grams are extremely small. Adding noises to their small counts can produce too many meaningless trajectories.

We address these challenges by generalizing the location unverse at each time point, i.e., merging locations based on trajectory distances in the original database. By doing so, the size of the location universe is greatly reduced, which further significantly reduces the size of the trajectory universe. In addition, by merging locations at each time point, many trajectories are also merged, which means their counts should be no longer too small. As a result, adding small noise cannot greatly affect the data utility.

Specifically, the proposed mechanism is composed of two key algorithms, both of which provides differential privacy guarantee:

(1) *Differentially Private Location Generalization Algorithm*: this algorithm probabilistically partitions the location universe Γ_i at each time point t_i and replaces all the locations belonging to the same group with their centroid. If we assume that the size of Γ_i is s and these locations are finally partitioned into m groups, then the total number of partition strategies is m^s. These candidates are probabilistically selected based on their score values, which are inversely proportional to Euclidean distances among trajectories within each group. This measure make this algorithm prefer to merging locations belonging to closer trajectories. This reduce the side effects on data utility. We make this process follow the exponential mechanism introduced earlier to enforce differential privacy. This step actually transforms the original location domain Γ_i to a more general one $\tilde{\Gamma}_i$ with much fewer locations.

(2) *Differentially Private Publishing Algorithm for Generalized Trajectories*: this algorithm generates new time-serials trajectories over the generalized location domains, and publishes their noisy counts based on the Laplace mechanism. This process can enforce differential privacy because it follows the Laplace mechanism. In addition, to guarantee the efficiency process, we take a special mechanism to avoid counting each possible trajectory over the generalized location domains. This measure can also make sure that the final output database $\tilde{\mathbb{D}}$ having the same size with $\mathbb{D}$, which is a requirement defined in Sec. III-C.

Because the above two algorithms provide differential privacy independently, the whole mechanism also provides differential privacy according to the sequential composition property of differential privacy. However, the privacy cost is the sum of that in the two algorithms.

Example. Consider the trajectory dataset containing eight trajectories in Fig. 1. As we show in Fig. 2, in the first algorithm, the proposed generalization algorithm partitions the locations at each time point into two groups and all the locations belonging to the same group are represented by a single point–their centroid. Then, in the second algorithm, it outputs the noisy counts of each trajectories formed by those generalized locations.

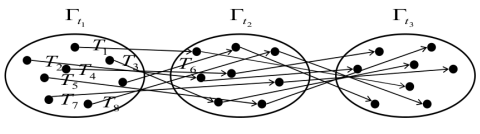
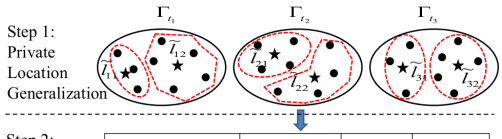


Fig. 1: A raw trajectory dataset including eight trajectories

We now elaborate the detailed design of the two major algorithms of the proposed mechanism.



Step 2: Private Release of Generalized

Generalized Tr.	Original Tr.	Real #	Noisy #
$\tilde{l}_{11} \rightarrow \tilde{l}_{21} \rightarrow \tilde{l}_{31}$	T_2	1	0
$\tilde{l}_{11} \rightarrow \tilde{l}_{21} \rightarrow \tilde{l}_{32}$	null	0	1
$\tilde{l}_{11} \rightarrow \tilde{l}_{22} \rightarrow \tilde{l}_{31}$	null	0	0
$\tilde{l}_{11} \rightarrow \tilde{l}_{22} \rightarrow \tilde{l}_{32}$	T_5, T_7	2	3
$\tilde{l}_{12} \rightarrow \tilde{l}_{21} \rightarrow \tilde{l}_{31}$	T_4, T_6	2	4
$\tilde{l}_{11} \rightarrow \tilde{l}_{21} \rightarrow \tilde{l}_{32}$	T_1, T_3	2	1
$\tilde{l}_{12} \rightarrow \tilde{l}_{22} \rightarrow \tilde{l}_{31}$	null	0	1
$\tilde{l}_{11} \rightarrow \tilde{l}_{22} \rightarrow \tilde{l}_{32}$	T_8	1	2

Fig. 2: Differentially private release of the sample data in Fig. 1

B. Differentially Private Location Generalization

We use an exponential mechanism introduced in Sec. III-B. to probabilistically partition the location domain at every time point. As we mentioned earlier, the total number of partition candidates is m^s for every time point $t_i \in Time(\mathbb{D})$, where s is the size of the original location domain Γ and m is the expected number of partitions. This candidate set is denoted by τ. We then define a utility function $U : \mathbb{D} \times \tau \rightarrow \mathbb{R}$ that assigns a score value for each candidate partition $p \in \tau$. Let LS_p^k denote the set of locations that are partitioned into group k according to the candidate partition p, and $D_{LS_p^k}$ denote the set of trajectories pass locations in LS_p^k at time t_i, namely, $D_{LS_p^k} = \{T : T(t_i) \in LS_p^k\}$. Let $AvgDist_p^k$ denote the average pairwise Euclidean distance among trajectories in $D_{LS_p^k}$, i.e.,

$$AvgDist_p^k = \frac{1}{\binom{2}{|D_{LS_p^k}|}} \sum_{T_i, T_j \in D_{LS_p^k}} Distance(T_i, T_j).$$

We further define $AvgDist(p) = \frac{1}{m} \sum_{k=1}^{m} AvgDist_p^k$. Then, the utility function is:

$$U(\mathbb{D}, p) = \frac{Min_{\rho \in \tau}(AvgDist(\rho))}{AvgDist(p)}. \quad (2)$$

The sensitivity of this function is 1 because the value of $U(\mathbb{D}, p)$ can vary at most 1 due to the change of a record. This utility function makes the proposed algorithm prefer to merging locations belonging to closer trajectories, which is beneficial for preserving data utility. Given the utility of all the candidate partitions, we can use the exponential mechanism to select one partition $p_i \in \tau$ following the probability:

$$\frac{exp(\frac{\epsilon_1}{2\Delta u} U(\mathbb{D}, p_i))}{\sum_{p \in \tau} exp(\frac{\epsilon_1}{2\Delta u} U(\mathbb{D}, p))}, \quad (3)$$

where Δu is the sensitivity of the utility function, i.e., equal to 1. According to Theorem 3, this process of location domain partition satisfies ϵ_1-differential privacy.

Although the above exponential scheme seems good in theory, it is non-trivial to apply it into practice. As we mentioned above, it has to compute and assign probabilities for all the m^s partition strategies, and then probabilistically picks one from them. This task may be computationally infeasible if m or s is extremely big. Such situation is almost inevitable for a huge trajectory database. For example, in the experimental database in this paper (Please find details on this database in IV-C .), the size of the original location universe at a single time point could reach 6000, namely, $s = 6000$. If we assume the locations are finally divided into 50 groups, the total number of partition strategies reaches 50^{6000}. It is infeasible to compute a probability for each of them based on Equation 3.

We thus propose an optimization to improve the efficiency. First of all, we use the classic k-means clustering to partition the original trajectories into m groups based on their pairwise Euclidean distances. This actually partitions the locations at each time point into m groups. We denote by this partition $\tilde{P}$. For such a partition, we have the following two important observations:

(1) Although we cannot guarantee that this partition is optimal, i.e., maximizes the utility function defined in Equation 2, it must approach the optimum since K-Means prefers to grouping closer trajectories.

(2) Adding or removing a single trajectory usually cannot significantly change the partition result. In other words, the output range of K-Means after adding or removing a single trajectory should be only a small subset of τ modified from $\tilde{P}$ instead of the whole τ.

We then leverage these two observations to redefine the utility function above. First, we revise Equation 2 based on the first observation. Denote by $\tilde{T}_i$ $(i = 1, 2, \cdots, m)$ the mean trajectory of the i-th group. The new function is

$$U(\mathbb{D}, p) = \frac{MeanDist(\tilde{P})}{MeanDist(p)}, \quad (4)$$

where

$$MeanDist(p) = \frac{1}{m \cdot |D_{LS_p^k}|} \sum_{k=1}^{m} \sum_{T_i \in D_{LS_p^k}} Distance(T_i, \tilde{T}_k).$$

According to the principle of k-means clustering, it is easy to obtain that $\forall p \in \tau \; MeanDist(p) \geq MeanDist(\tilde{P})$. Thereby, The sensitivity of this new function is still 1.

Second, we refine the output range τ of the exponential mechanism based on the second observation. Specifically, we define that τ, i.e., the set of partition candidates is formed by two parts:

- the k-means partition $\tilde{P}$ based on the original database $\mathbb{D}$, and another φ partitions producing the next φ greatest utilities, which are named φ-suboptimal partitions.
- a total of s k-means partitions based on the datasets, each of which removes a distinct trajectory from $\mathbb{D}$.

We present the algorithm to find the φ-suboptimal partitions in Algorithm 1. It outputs φ sets of membership modifications of trajectories from $\tilde{P}$ that produces the next φ greatest utilities. By doing so, the total number of partition candidates we have

to consider is reduced from m^s to $\varphi + 1 + s$, which makes the exponential scheme become feasible in practice.

Algorithm 1 $\varphi SubOptimal$

Input:
Raw Trajectory Database $\mathbb{D}$;
The optimal partition using k-means P_{Opt};
Output:
φ membership modifications of trajectories from the k-means partition result which cause the least utility loss;

$indvSubs = \varphi SubOptimalIndividual(\mathbb{D}, P_{Opt})$;
$result = \{\{indvSubs[0]\}\}$;
for $i = 1; i < sizeof(indvSubs)$ $i++$ **do**
  $a = indvSubs[i]$;
  $temp = \{\{a\}\}$;
  for each $mod \in result$ **do**
    if $\forall$ $elm \in mod$: $elm.T \neq a.T$ **then**
      $temp = temp \bigcup \{mod \cup a\}$;
    end if
    $result = result \bigcup temp$;
    Sort $result$ based on $sumDis_e$ of each element $e \in result$: $subDis_e = \sum_{i=0}^{sizeof(e)} e[i].Dis$;
    if $sizeof(result) > \varphi$ **then**
      $result = result[0, 1, \cdots, \varphi - 1]$;
    end if
  end for
end for
return $result$;

Procedure 2 $\varphi SubOptimalIndividual$

Input:
Raw Trajectory Database $\mathbb{D}$;
The optimal partition using k-means P_{Opt};
Output:
φ individual membership modifications of trajectories from the k-means partition result which cause the least utility loss;

Initialize $modifications = \{(T, G_k, Dis) | T \in \mathbb{D} \wedge T \notin \mathbb{D}_{P^k_{Opt}}, k = 1, 2, \cdots, m\}$;
// $Dis = Distance(T, \tilde{T}_{G_k}) - Distance(T, \tilde{T}_{G_{Opt}})$;
Sort elements of $modifications$ based on Dis_{TG_k};
return $modifications[0, 1, \cdots, \varphi - 1]$;

C. Differentially Private Release of Generalized Trajectories

By using the above algorithm, the original location universe Γ_i at each time point t_i is transformed to a generalized universe $\tilde{\Gamma}_i$. As a result, the real counts of trajectories should have been greatly improved. The task of the second algorithm is to produce generalized trajectories over those new location universes.

Let Ω denote the universe of trajectories with locations drawn from the generalized location universes. The naive way to guarantee ϵ_2-differential privacy in this task is to explicitly consider all the trajectories in Ω and output their noisy counts based on the Laplace mechanism, i.e., add noise $Lap(1/\epsilon_2)$ to each true count. Unfortunately, this is still infeasible due to the extremely high dimension of trajectory data even if the location universes have been greatly compressed in the first step. Let's use the experimental database in this paper as an example again. Suppose that the location universe at each time point is partitioned into 50 groups, i.e., each universe contains 50 locations, the total number of possible trajectories of length 32 that we have to consider is 50^{32}, which is obviously uncomputable with today's system.

Our solution to this challenge consists of the following two steps

First, we count the trajectories generalized from those in the original database, and add Laplace noise $Lap(1/\epsilon_2)$ to each of the true counts. Let $\tilde{\mathbb{D}} = \{\tilde{T}_1, \tilde{T}_2, \cdots, \tilde{T}_k\}$ denote the set of such trajectories ordered by their noisy counts, namely, $C_1 > C_2 > \cdots > C_k$, where C_i represents the noisy count of $\tilde{T}_i \in \tilde{\mathbb{D}}$.

Second, beginning with $(C_2, C_1]$, we compute the expected number Num_i of trajectories in $\Omega - \tilde{\mathbb{D}}$ whose noisy counts are located in $(C_{i+1}, C_i]$ according to the naive mechanism. Let $f(x, \epsilon_2)$ denote the density function of the Laplace distribution. The true count of every trajectory in $\Omega - \tilde{\mathbb{D}}$ is 0. After adding Laplace noise $Lap(1/\epsilon_2)$, the probability that a trajectory in $\Omega - \tilde{\mathbb{D}}$ owns a count within $(C_{i+1}, C_i]$ is $\int_{C_{i+1}}^{C_i} f(x, \epsilon_2) dx$. Therefore, it is easy to derive that

$$Num_i = |\Omega - \tilde{\mathbb{D}}| \cdot \int_{C_{i+1}}^{C_i} f(x, \epsilon_2) dx. \quad (5)$$

We randomly pick this number of unique trajectories from $\Omega - \tilde{\mathbb{D}}$ and include them as well as $\tilde{T}_i$ into the final output set. Their noisy counts are random values within this interval. This process stops when the total counts of the trajectories in the output set reaches the size of the original dataset $\mathbb{D}$.

We finally release the trajectories in the final output set as well as the corresponding noisy counts. The counts of all the other trajectories in Ω are ignored and denoted by the same symbol ϕ.

In the above design, we ignore a fact that the distance between two continuous locations in a trajectory is bounded by the moving speed of the user. In particular, if we assume that the moving speed of the user is c, the distance $Dist_{l_i, l_{i+1}}$ between the locations l_i and l_{i+1} of T should satisfy $Dist_{l_i, l_{i+1}} \leq c \times (t_{i+1} - t_i)$. Considering this constraint, we can significantly optimize our algorithm:

(1) The total number of possible trajectories can be greatly reduced. As a result, we will produce fewer noisy trajectories in the second step. We propose an $\bigcirc(|T||\tilde{\Gamma}|^2)$ algorithm to count this number. Due to the space limitation, we ignore this algorithm here. Reducing noisy trajectories will certainly increase the data utility.

(2) When we randomly pick noisy trajectories, we should also consider this constraint to avoid producing meaningless trajectories that can be easily filtered out.

V. Privacy Analysis

We now analyze the privacy guarantee of the proposed mechanism.

This mechanism is composed of two key algorithms. The first one leverages an exponential mechanism to probabilistically generalize the location domain at every time point.

According to Theorem 3, each of such generlizarion is ϵ_1-differentially private. Moreover, as these partitions are sequential, we have the following lemma based on the sequential composition rule of Theorem 4:

Lemma 1. *The location generalization algorithm proposed in Sec. IV-B is ($|T| \cdot \epsilon_1$)-differentially private.*

The second algorithm utilizes the Laplace mechanism to produce noisy counts of generalized trajectories. We now try to prove the following lemma.

Lemma 2. *The trajectory publishing algorithm proposed in Sec. IV-C is ϵ_2-differentially private.*

Proof. Suppose that $\mathbb{D}$ is a trajectory database differing in one trajectory, which is denoted by T_x, from $\mathbb{D}$. Let $\tilde{T}_x \in \Omega$ be the generalized trajectory of T_x. We write $NC^i_\Omega(\mathbb{D})$ as the noisy count of the generalized trajectory $\tilde{T}_i \in \Omega$ based on the original database $\mathbb{D}$. The probability of output $r = \{r_1, r_2, \cdots, r_{|\Omega|}\}$ from the sequence of $NC^i_\Omega(\mathbb{D})$ is

$$Prob[NC_\Omega(\mathbb{D}) = r] = \prod_i Prob[NC^i_\Omega(\mathbb{D}) = r_i]$$

It is easy to find that $\forall \tilde{T}_i \in \Omega \wedge \tilde{T}_i \neq \tilde{T}_x$, $Prob[NC^i_\Omega(\mathbb{D}) = r_i] = Prob[NC^i_\Omega(\mathbb{D}') = r_i]$. When $\tilde{T}_i = \tilde{T}_x$, there are three subcases:

Case 1 $\tilde{T}_x \in \tilde{\mathbb{D}}$: In this case, the noise count of $\tilde{T}_x$ is obtained by adding Laplace noise $Lap(1/\epsilon_2)$ to its real count. Thereby, $Prob[NC^x_\Omega(\mathbb{D}') = r_x] \leq Prob[NC^x_\Omega(\mathbb{D}) = r_x] \times e^{\epsilon_2}$.

Case 2 $\tilde{T}_x \notin \tilde{\mathbb{D}} \wedge r_x \neq \phi$: We assume that $r_x \in (C_i + 1, C_i)$. Based on Equation 5, we can derive that

$$\begin{aligned} Prob[NC^x_\Omega(\mathbb{D}') = r_x] &= \frac{1}{C_i - C_{i+1}} \cdot \int_{C_{i+1}-1}^{C_i - 1} f(x, \epsilon_2) dx \\ &= \frac{1}{2(C_i - C_{i+1})} \cdot e^{\epsilon_2} \cdot \\ &\quad [e^{-\epsilon_2 C_{i+1}} - e^{-\epsilon_2 C_i}] \\ &= e^{\epsilon_2} \cdot Prob[NC^x_\Omega(\mathbb{D}) = r_x]. \end{aligned}$$

Case 3 $\tilde{T}_x \notin \tilde{\mathbb{D}} \wedge r_x = \phi$: Let C_{min} denote the minimal noisy count of trajectories included in the output set. It is easy to derive that

$$\begin{aligned} Prob[NC^x_\Omega(\mathbb{D}') = \phi] &= 1 - \int_{C_{min}-1}^{C_1 - 1} f(x, \epsilon_2) dx \\ &= 1 - \frac{1}{2} \cdot e^{\epsilon_2} \cdot [e^{-\epsilon_2 C_{min}} - e^{-\epsilon_2 C_1}] \\ &\leq e^{\epsilon_2} - \frac{1}{2} \cdot e^{\epsilon_2} \cdot [e^{-\epsilon_2 C_{min}} - e^{-\epsilon_2 C_1}] \\ &= e^{\epsilon_2} \cdot Prob[NC^x_\Omega(\mathbb{D}) = \phi]. \end{aligned}$$

Based on the analysis above, we obtain $Prob[NC_\Omega(\mathbb{D}) = r] = e^{\epsilon_2} \cdot Prob[NC_\Omega(\mathbb{D}') = r]$. Thus, according to Definition 1, we can conclude that this algorithm satisfies ϵ_2-differential privacy. □

As the first and the second algorithms are sequential, we can further derive the following theorem based on Theorem 4:

Theorem 5. *The proposed algorithm satisfies ϵ-differentially private, where $\epsilon = |T| \cdot \epsilon_1 + \epsilon_2$.*

VI. Evaluation

In this section, we evaluate the performance of the propose mechanism in anonymizing a real trajectory dataset. We focus on two important measurements in the evaluation: the utility of the published data and the time efficiency of the anonymization process.

A. Experiment setup

The data used in our experiment is from a real trajectory database, which consists of the trajectory data of 10357 taxis collected by Microsoft Research within one week at Beijing, China. Every node in a trajectory is composed of the taxi ID, the recording time, and the current location (i.e., the latitude and longitude). Although all of these trajectories are recorded within the same week, their detailed time periods are very different. Thus, we choose the trajectories within the same period from 8 : 30 to 14 : 30 as our experimental data. In addition, each trajectory is refined to contain 32 nodes, in which the interval between any two adjacent ones is 10 minutes. After the pre-processing, we obtain a total of 6013 trajectories. Note that all the distance shown below is the Euclidean distance computed over the latitude/longitude coordinates.

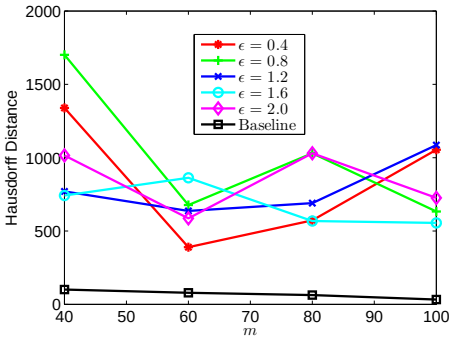
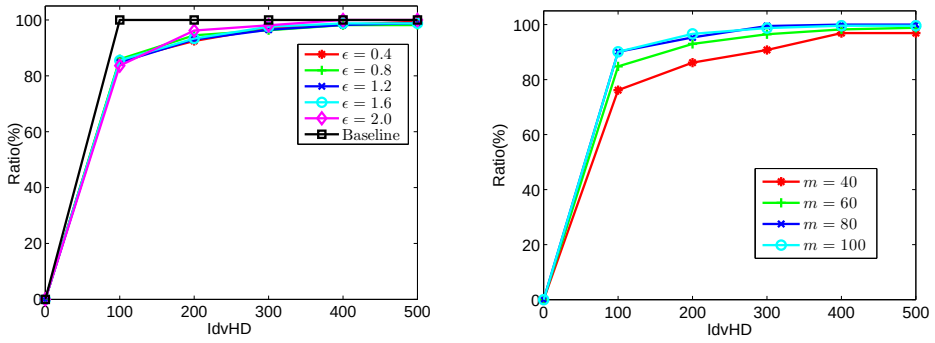


Fig. 3: Hausdorff distances of the published data

B. Utility evaluation

In the propose mechanism, both the key algorithms bring side effects on the data utility. The first one probabilistically merges the location nodes in trajectories, which will distort the original trajectories. The second one adds noise trajectories to the final output while deletes some real ones. In this subsection, we first evaluate the utility losses in terms of Hausdorff distance (See Sec. 1) between $\mathbb{D}$ and $\tilde{\mathbb{D}}$. In addition, as the purpose of publishing data is to query or analyze it, we further measure the utility by comparing the results of real queries on $\mathbb{D}$ and $\tilde{\mathbb{D}}$. In particular, we adopt *spatio-temporal range queries with uncertain*, which is proposed by Trajcevski et al. [20] to query moving objects inside a spatio-temporal range according to their uncertain trajectories. Abul et al. [1]



(a) $m = 60$ (b) $\epsilon = 1.2$

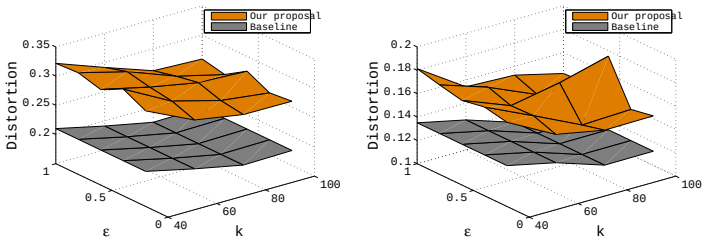
Fig. 4: CDF of the distribution of individual Hausdorff distances

first apply such queries to evaluate the utility of trajectory data sanitized by their *NWA* mechanism.

1) Hausdorff distance: In both evaluations, we use NWA [1], which is one of the classical k-anonymity based trajectory publishing mechanism as the baseline. This method proposes the concept of (k, δ)-anonymity, which exploits the uncertainty of moving objects to relax the restriction of traditional k-anonymity and thus reduces the amount of utility losses due to anonymization. Here, δ represents the position error. We set it to 10 in this paper. We adopt NWA as the baseline since it also needs to cluster the original trajectories based on the Euclidean distance between each other.

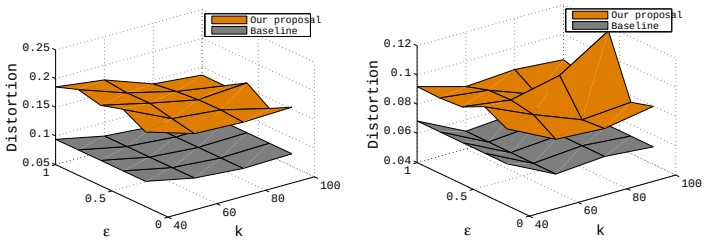
We first measure the data utility of the data published by NWA and our proposal with the Hausdorff distances (defined in Equation 1). We find that the Hausdorff distance of the data published by our method is much higher than that of the data published by NWA. This is reasonable since the Hausdorff distance is determined by the worst record. In our proposal, the published data includes a set of noisy trajectories that do not exist in the original data, which must significantly increase the Hausdorff distance. NWA does not include such noisy trajectories in the published data and thus its Hausdroff distance is much smaller.

Nevertheless, this does not mean that the real utility of our data is as worse as the Hausdorff distance. Let's consider the distribution of Hausdroff distances for individual trajectories. For a specific trajectory $T \in \tilde{\mathbb{D}}$, we define $IdvHD(T) = min_{T' \in \mathbb{D}} Distance(T, T')$. We present the CDF of the distribution of $IdvHD(T)$, $T \in \tilde{\mathbb{D}}, \mathbb{D})$, in Fig. 4. The results show that more than 80% of trajectories in our approach have similar utility with those in NWA. This indicates that the results of queries or analyses based on our data should not be much worse than those based on NWA data, which has been demonstrated by our following evaluation based on spatio-temporal range queries. In addition, the utility increases as m increases. This is reasonable since a larger m predicts that every location will merge with fewer other locations. As a result, the average loss of location precision is smaller.



(a) Radius=0.5 (b) Radius=1

Fig. 5: Range query distortion for Q_1



(a) Radius=0.5 (b) Radius=1

Fig. 6: Range query distortion for Q_2

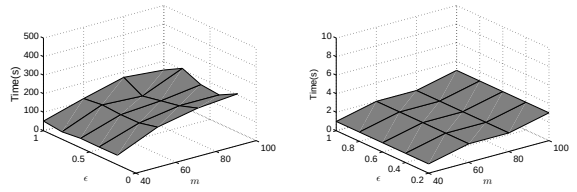
2) Spatio-temporal range queries: The same as in the paper of NWA[1], we consider the following two kinds of spatio-temporal range queries,

- Q_1: select count(*) from D where $PSI(T \in D, R, t_b, t_e)$
- Q_2: select count(*) from D where $DAI(T \in D, R, t_b, t_e)$

Here, the functions PSI (*Possibly_Sometime_Inside*) and DAI (*Definitely_Always_Inside*)are formally defined below:

$$PSI(T, R, t_b, t_e) = (\exists f_{PMC^T})(\exists t \in [t_b, t_e])inside(R, f_{PMC^T}(t), t),$$

$$DAI(T, R, t_b, t_e) = (\forall f_{PMC^T})(\forall t \in [t_b, t_e])inside(R, f_{PMC^T}(t), t).$$



(a) Total time costs

(b) Time costs excepting those for clustering

Fig. 7: Time efficiency

The function f_{PMC^T}: $Time \rightarrow \mathbb{R}^2$ is a continues function such that $f_{PMC^T}(t)$ represents the set of spatio-temporal points inside the uncertainty area of T at time t. The notion R is a region determined by a central point and a radius, and t_b and t_e define a time period.

We use the *range query distortion*, i.e., $\frac{|Q(\mathbb{D})-Q(\tilde{\mathbb{D}})|}{max(Q(\mathbb{D}),Q(\tilde{\mathbb{D}}))}$, as a metric to evaluate the utility of the published data $\tilde{\mathbb{D}}$. We vary the privacy budget ϵ and the partition number m of the location domain, and average the results over 1000 runs with randomly selected regions having three different radiuses. The results are presented in Fig. 5. We can see that the distortions introduced by our mechanism are all blow the twices of those introduced by NWA. Given that our mechanism guarantee a more strict privacy model, we think such results are affordable. In addition, we find that this difference decreases as the region size increases.

C. Efficiency

This experiment is realized by java and conducted on a laptop with intel i7-3840QM processor and a 16G memory. The total time is shown in Fig. 7a. We can find the time cost increases with m, but changes slightly with ϵ. Actually, as you can find from Fig. 7b that the time cost is dominated by the clustering work. After excluding this part, the remained time cost is less than 5s, which increases slightly with m.

VII. Conclusion

In this paper, we have proposed a novel approach for differentially private publishing general time-serial trajectory data. It does not make an implicit assumption that the trajectories to be published contain a lot of identical prefixes or n-grams, which is not always true in the real world. This mechanism first uses an exponential mechanism to probabilistically merge locations at each time point based on the trajectory distances, and then leverages a carefully designed noisy counting scheme adapted from the traditional Laplace mechanism to release a set of generalized trajectories with the same size with the raw dataset. We adopt special techniques in both steps to avoid exhaustively accounting each candidate in the output domain to guarantee the efficiency. Extensive experiments on real-life datasets proved that although our solution provides stronger privacy protection than those just guarantee k-anonymity, the side effects on data utility are not significantly increased.

References

[1] Osman Abul, Francesco Bonchi, and Mirco Nanni. Never walk alone: Uncertainty for anonymity in moving objects databases. In *ICDE*, pages 376–385, 2008.

[2] Xin Cao, Gao Cong, and Christian S. Jensen. Mining significant semantic locations from gps data. *PVLDB*, 3(1):1009–1020, 2010.

[3] Rui Chen, Gergely Ács, and Claude Castelluccia. Differentially private sequential data publication via variable-length n-grams. In *ACM Conference on Computer and Communications Security*, pages 638–649, 2012.

[4] Rui Chen, Benjamin C. M. Fung, and Bipin C. Desai. Differentially private trajectory data publication. *CoRR*, abs/1112.2020, 2011.

[5] Rui Chen, Benjamin C. M. Fung, Noman Mohammed, Bipin C. Desai, and Ke Wang. Privacy-preserving trajectory data publishing by local suppression. *Inf. Sci.*, 231:83–97, 2013.

[6] Roger Clarke. Person location and person tracking - technologies, risks and policy implications. *IT & People*, 14(2):206–231, 2001.

[7] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284, 2006.

[8] Srivatsava Ranjit Ganta, Shiva Prasad Kasiviswanathan, and Adam Smith. Composition attacks and auxiliary information in data privacy. In *KDD*, pages 265–273, 2008.

[9] Haibo Hu, Jianliang Xu, Sai Tung On, Jing Du, and Joseph Kee-Yin Ng. Privacy-aware location data publishing. *ACM Trans. Database Syst.*, 35(3), 2010.

[10] Kaifeng Jiang, Dongxu Shao, Stéphane Bressan, Thomas Kister, and Kian-Lee Tan. Publishing trajectories with differential privacy guarantees. In *SSDBM*, page 12, 2013.

[11] Daniel Kifer. Attacks on privacy and definetti's theorem. In *SIGMOD Conference*, pages 127–138, 2009.

[12] Frank McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *SIGMOD Conference*, pages 19–30, 2009.

[13] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *FOCS*, pages 94–103, 2007.

[14] Noman Mohammed, Rui Chen, Benjamin C. M. Fung, and Philip S. Yu. Differentially private data release for data mining. In *KDD*, pages 493–501, 2011.

[15] Anna Monreale, Gennady L. Andrienko, Natalia V. Andrienko, Fosca Giannotti, Dino Pedreschi, Salvatore Rinzivillo, and Stefan Wrobel. Movement data anonymity through generalization. *Transactions on Data Privacy*, 3(2):91–121, 2010.

[16] Marie-Pier Pelletier, Martin Trepanier, and Catherine Morency. Smart card data use in public transit: A literature review. *Transportation Research C: Emerging Technologies*, 19(4):557–568, 2011.

[17] Dongxu Shao, Kaifeng Jiang, Thomas Kister, Stéphane Bressan, and Kian-Lee Tan. Publishing trajectory with differential privacy: A priori vs. a posteriori sampling mechanisms. In *DEXA (1)*, pages 357–365, 2013.

[18] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.

[19] Manolis Terrovitis and Nikos Mamoulis. Privacy preservation in the publication of trajectories. In *MDM*, pages 65–72, 2008.

[20] Goce Trajcevski, Ouri Wolfson, Klaus Hinrichs, and Sam Chamberlain. Managing uncertainty in moving objects databases. *ACM Trans. Database Syst.*, 29(3):463–507, 2004.

[21] Raymond Chi-Wing Wong, Ada Wai-Chee Fu, Ke Wang, Philip S. Yu, and Jian Pei. Can the utility of anonymized data be used for privacy breaches? *TKDD*, 5(3):16, 2011.

[22] Roman Yarovoy, Francesco Bonchi, Laks V. S. Lakshmanan, and Wendy Hui Wang. Anonymizing moving objects: how to hide a mob in a crowd? In *EDBT*, pages 72–83, 2009.

[23] Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. Mining interesting locations and travel sequences from gps trajectories. In *WWW*, pages 791–800, 2009.