THE NATIONAL UNIVERSITY
of SINGAPORE

School of Computing
Computing 1, 13 Computing Drive, Singapore 117417

TRA4/13

Publishing Trajectory with Differential Privacy: A Priori vs A Posteriori Sampling Mechanisms

Dongxu Shao, Kaifeng Jiang, Thomas Kister, Stephane Bressan and Kian-Lee Tan

April 2013

T e c h n i c a l R e p o r t

Foreword

This technical report contains a research paper, development or tutorial article, which has been submitted for publication in a journal or for consideration by the commissioning organization. The report represents the ideas of its author, and should not be taken as the official views of the School or the University. Any discussion of the content of the report should be sent to the author, at the address shown on the cover.

OOI Beng Chin
Dean of School

Publishing Trajectory with Differential Privacy: A Priori vs A Posteriori Sampling Mechanisms

Dongxu Shao [1], Kaifeng Jiang [2], Thomas Kister [1], Stéphane Bressan [2], and Kian-Lee Tan [1]

[1]School of Computing, National University of Singapore
{shaodx,thomas.kister,tankl}@comp.nus.edu.sg
[2]Center for Maritime Studies, National University of Singapore
{cmsjk,steph}@nus.edu.sg

Abstract. It is now possible to collect and share trajectory data for any ship in the world by various means such as satellite and VHF systems. However, the publication of such data also creates new risks for privacy breach with consequences on the security and liability of the stakeholders. Thus, there is an urgent need to develop methods for preserving the privacy of published trajectory data. In this paper, we propose and comparatively investigate two mechanisms for the publication of the trajectory of individual ships under differential privacy guarantees. Traditionally, privacy and differential privacy is achieved by perturbation of the result or the data according to the sensitivity of the query. Our approach, instead, combines sampling and interpolation. We present and compare two techniques in which we sample and interpolate (a priori) and interpolate and sample (a posteriori), respectively. We show that both techniques achieve a $(0, \delta)$ form of differential privacy. We analytically and empirically, with real ship trajectories, study the privacy guarantee and utility of the methods.

1 Introduction

With the increasing pervasiveness of high quality location-acquisition technologies, geolocation becomes the bread and butter of many applications. In those applications traditionally concerned with navigation such as shipping, new analytical and operational opportunities are created. However, the possibility to collect and share trajectory data for any ship in the world by various means such as satellite and VHF systems creates new risks for privacy breach with detrimental consequences on the security and liability of the stakeholders. For example, pirates can use such data to track the frequent routes of ships, and lay ambush to launch attacks. For this reason, the International Maritime Organization (IMO) has warned against the publication of ship trajectory[1]. Moreover, trajectory data generally contain sensitive information [2]. Any improper publication of such sensitive data can lead to privacy breach. In fact, as every position is potentially

[1] http://www.imo.org/ourwork/safety/navigation/pages/ais.aspx

sensitive, it is critical to protect the privacy of each individual position in the trajectory. This motivates us to investigate the problem of publishing trajectory data with differential privacy.

ε-differential privacy was first introduced by Dwork in 2006 [5], and it is now a widely accepted privacy standard. It requires that the output answer by the randomized mechanism to a query function be insensitive to any change of a single element in the underlying database. The insensitivity is controlled by the parameter ε. For example, a ship passes a specific position for some business purpose which should not be revealed to public. Meanwhile, its whole trajectory is published by some website (e.g. [12]). If the publication were ε-differentially private, then it is of high probability that the published trajectory is the same as that created from any trajectory excluding this position. Hence it is very difficult for attackers to obtain truthful information about any position by analyzing the published trajectories.

The first method for achieving differential privacy is the Laplace mechanism [7], which adds random noise following the Laplace distribution to the true answers to the query functions. The level of Laplace noise needs to be calibrated to the sensitivity of the query function. Another solution for differential privacy is exponential mechanism, proposed by McSherry and Talwar [16]. This mechanism works for all kinds of data sets, such as strings, strategies, or trees. Both of the two mechanisms perturb the results according to the sensitivity of the query. The query in our consideration here is the trajectory itself. Its sensitivity is very high because the velocity of a ship can be very fast. Hence it is very hard to get good utility by using these two common methods. In order to obtain reasonable utility, we adopt a relaxed version of ε-differential privacy.

Dwork et al. [6] proposed (ε, δ)-differential privacy, where δ bounds the probability that ε-differential privacy does not happen. In this paper, we propose two mechanisms using combination of sampling and interpolation to preserve $(0, \delta)$-differential privacy. Therefore, our proposal guarantee that the strongest version of ε-differential privacy happens except for a little probability δ. This privacy preserving is obtained by the sampling stage. The interpolation stage is designed to deliver trajectories with good utility. These two stages can be in any order. We also compare a priori sampling mechanism and a posteriori sampling mechanism.

Contribution: In this paper, we consider the problem of publishing trajectories via the differential privacy model. The key challenge is to improve the utility of the mechanism while preserving privacy level. Our proposed mechanisms are able to achieve the strongest differential privacy except a small probability. We comparatively evaluate the performance of this mechanism both qualitatively by illustrating the publication of real ship trajectories and quantitatively by measuring the error between the published and original trajectory.

- We propose a priori sampling mechanism (**SFI**[2]) and a posteriori sampling mechanism (**IFS**[3]) to publish trajectories with $(0, \delta)$-differential privacy.
- We compare these two mechanisms analytically and empirically.

[2] SFI stands for Sampling First and Interpolation.
[3] IFS stands for Interpolation First and Sampling.

– We conduct numerical experiments to evaluate the utilities of **SFI** and **IFS**. The numerical results show that the **SFI** mechanism has better performance.

The rest of this paper is organized as follows. Section 2 reviews related work. Formal definitions and problem statement are introduced in Section 3. We present our two sampling-based differentially private mechanisms in Section 4. We report the numerical results in Section 5 and give the final conclusion in Section 6.

2 Related Work

Generally there are mainly two different types of trajectory publishing. One type aims to publish a group of trajectories and considers each trajectory as one individual record. The other type considers one trajectory as a database and each position in the trajectory as one individual record.

Recent privacy-preserving technology for the first type starts with the concept (k, δ)-anonymity proposed by Abul et al.[1]. The intuition is to disturb the trajectory so that at least k many different trajectories co-exist in a cylinder with radius δ. Chen et al.[4] were among the first to connect trajectory publishing and differential privacy. They proposed a data-dependent sanitization mechanism by building a noisy prefix tree according to the underlying data. The most recent work by Ho[11] proposed a differential privacy mechanism for mining trajectory data. The approach adds Laplace noise with respect to the smooth sensitivity of queries.

To our knowledge, not much work has been done on the second type which treats each individual trajectory as a database. In fact, to a vehicle owner, every position of the trajectory could be potentially sensitive. The privacy of each position can be preserved by sampling. This is the focus of our paper. Besides, we use interpolation to retrieve the information of sampled-out positions.

The interpolation method in our proposal is a standard and classic one, which is widely used in robot route planning. A simple task is to find smooth enough paths passing through a sequence of given waypoints. Sometimes additional requirements on velocity and direction at waypoints are involved. A classic technology to achieve this is by using Bézier spline. To obtain a continuous curve matching given direction and velocity at waypoints, cubic Bézier was applied in [15], [18], [10]. Cubic Bézier curve requires two control points exclusive of endpoints and can keep continuous curvature from the beginning to the end.

This application of cubic Bézier spline fits our purpose of interpolation very well. We employ this method and combine it with sampling to publish privacy-preserving trajectories. In many applications, the data to be sanitized are collected via a simple random sampling from the underlying population.

The random sampling method would allow others to study the statistical patterns about the entire population based upon the collected sample data, e.g., averages, variances, clusters etc. Intuitively, a simple random sampling already provides certain amount of privacy guarantees for the underlying population. In

[3], Chaudhuri and Mishra have shown that a simple random sampling mechanism (without any further sanitization) does not preserve ε-differential privacy, and under certain conditions it may guarantee the probabilistic differential privacy that the ε-differential privacy is preserved with probability at least $1 - \delta$. These results were further modified and extended in [13]. In [17], in order to add smaller amount of instance-based noise than the worst-case noise determined by the global sensitivity, Nissim et al. proposed a sample and aggregate framework by replacing the query function with a related function whose smooth sensitivity is low and efficiently computable. In [14], Li et al. proved that applying random sampling, as a pre-sampling step, to k-anonymity methods can preserve (ε, δ)-differential privacy. In [9], Gehrke et al. introduced a new definition of privacy called crowd-blending privacy, which is a relaxation of differential privacy. The authors show that the crowd-blending mechanism, with a pre-sampling from the underlying population, can both guarantee differential privacy and the stronger notion of zero-knowledge privacy.

In the above mentioned random sampling results, only the sampled data would be released to public for statistical studies. However, in order to monitor the ship's navigation, we still would like to estimate the ship's possible positions in the time interval between any two sampled positions. Significant events, including illegal dumping of waste materials and oil spill, may happen in some time interval. Hence it is of great importance to infer the ship's positions during the navigation. Our proposal in this paper can achieve $(0, \delta)$-differential privacy for small δ. Moreover, a large number of experiments conducted on real ship trajectories demonstrate good utility of our mechanisms.

3 Preliminaries

In this section, we present the formal definition of differential privacy and Bézier curve, ending with the model of our problem.

3.1 Differential Privacy

Differential privacy has been widely used to protect the privacy of the individual participants while providing useful statistical information about the underlying population. A mechanism satisfies differential privacy if the addition or removal of a single database element does not significantly change the probability of any outcome of the mechanism.

Definition 1 (ε-differential privacy [5, 7]) *A randomized mechanism $\mathcal{K}$ gives ε-differential privacy if for every two databases D and D' differing in at most one row, and for every $S \subseteq \mathrm{Range}(\mathcal{K})$*

$$\Pr[\mathcal{K}(D) \in S] \leq e^{\varepsilon} \times \Pr[\mathcal{K}(D') \in S].$$

Dwork et al. [6] proposed (ε, δ)-differential privacy, which is a relaxed version of ε-differential privacy that allows privacy breaches to occur with a very small probability controlled by δ.

Definition 2 ((ε, δ)-Differential privacy[6]) *A randomized mechanism $\mathcal{K}$ gives the (ε, δ)-differential privacy if for every two databases D and D' differing in at most one row, and for every $S \subseteq \text{Range}(\mathcal{K})$,*

$$\Pr[\mathcal{K}(D) \in S] \leq \exp(\varepsilon) \times \Pr[\mathcal{K}(D') \in S] + \delta.$$

Note that if $\delta = 0$, $(\varepsilon, 0)$-differential privacy is ε-differential privacy.

Remark 1 *From the definition, we can see that the strongest ε-differential privacy happens when $\varepsilon = 0$. Normally this level of privacy is very difficult to achieve with reasonable utility. However, it may happen for certain probability. This is what $(0, \delta)$-differential privacy mean, where the probability of privacy breach is no greater than the constant δ. Consequently, $(0, \delta)$-differential privacy is meaningless when $\delta = 1$. Meanwhile, in the definition of ε-differential privacy, the privacy breach is measured by the ratio between the probabilities of outputting the same result with two neighboring inputs. Generally it is unfair to compare their privacy strengths.*

3.2 Bézier Curve

A Bézier curve is a smooth curve determined by a sequence of control points. Suppose there are $(n + 1)$ points $P_0, ..., P_n$. The first and last points are the endpoints of the curve. The other points are attractors of the curve. In other words, the curve should be close to these points, but generally does not pass through them.

Linear Bézier curves In the case of $n = 1$, the Bézier curve is just a straight line from P_0 to P_1:

$$B(t) = (1 - t)P_0 + t \cdot P_1, \ t \in [0, 1].$$

Quadratic Bézier curves In this case, we are given three points: P_0, P_1 and P_2. Then we have two linear interpolations for P_0, P_1 and P_1, P_2, respectively. The Bézier curve for these three points is a linear interpolation of these two linear interpolations as follows, $B(t) = (1 - t)[(1 - t)P_0 + tP_1] + t[(1 - t)P_1 + tP_2], \ t \in [0, 1]$.

A simple rearrangement gives us

$$B(t) = (1 - t)^2 P_0 + 2(1 - t)tP_1 + t^2 P_2, \ t \in [0, 1].$$

The derivative of this Bézier curve is

$$B'(t) = 2(1 - t)(P_1 - P_0) + 2t(P_2 - P_1).$$

This means that the direction at the starting point P_0 is heading to P_1, and the direction at the ending point P_2 is from P_1.

Generalization The Bézier curve for an arbitrary n can be constructed by recursion. Suppose there are $(n+2)$ points $P_0, ..., P_{n+1}$. Let $B_{P_0 \dots P_n}$ denote the Bézier curve for $P_0, ..., P_n$. Then the Bézier curve for $P_0, ..., P_{n+1}$ is

$$B(t) = (1-t)B_{P_0 \dots P_n}(t) + tB_{P_1 \dots P_{n+1}}(t),\ t \in [0,1].$$

3.3 Problem Statement

The simplest form of a trajectory is a sequence is positions on the 2-dimensional plane representing the moving path of a vehicle. To make our model more realistic, we can aggregate information as much as we want. In this paper, we consider trajectories with the direction, velocity and timestamp at every position. For simplicity, we assume that the information of the starting position and the terminal position is known to public.

Definition 3 *A trajectory T is a sequence $\langle (P_0, \theta_0, v_0, t_0), \dots, (P_n, \theta_n, v_n, t_n) \rangle$, where P_i is the coordinate of the i-th position, θ_i is the direction and v_i is the velocity.*

After we deliver an output for an input trajectory, we have to measure the utility of our delivery. There are many ways to measure the distance of two trajectories, based on different intuitions. Two measures are adopted here for two purposes.

Definition 4 *Given two trajectories T and $\widetilde{T}$, the MAX distance between them is*

$$MAX(T, \widetilde{T}) = \max\{||P_i - \widetilde{P}_i|| :\ 0 \le i \le n\}.$$

The MAX distance measures the maximum of the distance between positions with the same timestamp. Consequently, given the output, a timestamp t_i and the MAX distance r, it is guaranteed the real position P_i is in the circle centered at $\widetilde{P}_i$ with radius r. To calculate the MAX distance, the two trajectories must have the same timestamps. Since the goal of this paper is to publish a perturbed trajectory $\widetilde{T}$ while preserving $(0, \delta)$-differential privacy, we have to define neighboring trajectories to be with the same timestamp sequence.

Definition 5 *Two trajectories T and T' are neighboring if they have the same timestamp sequence and differ at exactly one tuple.*

Alternatively, one may be interested in the similarity between T and $\widetilde{T}$. The Dynamic Time Wapping distance (DTW) is an ideal to fulfill this task. The Dynamic Time Warping (DTW) algorithm defined recursively as:

$$\mathrm{DTW}(i,j) = \begin{cases} 0 & \text{if } i = -1 \text{ and } j = -1, \\ +\infty & \text{else if } i = -1 \text{ or } j = -1, \\ dist(P_i, \widetilde{P}_j) + min(\ \mathrm{DTW}(i-1, j), & \text{otherwise} \\ \qquad\qquad\qquad\quad \mathrm{DTW}(i, j-1), & \\ \qquad\qquad\qquad\quad \mathrm{DTW}(i-1, j-1)) & \end{cases}$$

where $dist(P_i, \widetilde{P}_j)$ is the cost function between the two points. We choose to define that function as the Euclidean distance between P_i and $\widetilde{P}_j$. The algorithm consists in walking along both trajectories, pairing points between the both of them, but allowing that for the next step only one of the trajectories is walked to its next point. Therefore a point can be paired with one or more consecutive points on the other trajectory. This allows us to measure the similarity between both trajectories' pattern.

4 Sampling-based Differentially Private Schemes

In this section, we shall present two differentially private schemes to protect an individual ship trajectory. Both schemes are based on sampling and interpolation. Our first scheme is an Apriori Sampling Scheme which first draws a sample from the trajectory data (i.e., by sampling each position with a given probability) and then applies interpolation to construct the published trajectory over the sampled data points. The second scheme, known as Aposteriori Sampling Scheme, first interpolates a smooth curve over the trajectory data points, and then sample points on this curve. In the following, we shall first describe the interpolation mechanism, then present the two mechanisms.

To analyze the privacy of our mechanisms, we first recall the composition lemma proved by Dwork et al. [8],

Lemma 1 *The class of (ε, δ)-differentially private algorithms satisfies $(k\varepsilon, k\delta)$-differential privacy under k-fold adaptive composition.*

As we shall see, the interpolation method we are using is deterministic. Hence, the privacy is fully taken care of by the sampling stage.

4.1 Interpolation

The aim of interpolation is to recover the information of sampled-out positions. Therefore, a well-chosen interpolation method would improve the utility. A classic method of route planning is employed here.

Suppose we have the information for starting position (P_s, θ_s, v_s) and the end position (P_e, θ_e, v_e). Suppose the length of time interval is t^*. To match the direction at P_s, we set an additional control point P_1 along the direction of θ_s with distance l_s from P_s. Similarly, another additional control point P_2 is chosen for P_e. These four points will form a cubic Bézier curve which automatically matches the directions at these two endpoints. Moreover, to match the velocity, it must be the case that $l_s = \dfrac{v_s \cdot t^*}{3}$ and $l_e = \dfrac{v_e \cdot t^*}{3}$. Hence,

$$P_1 = P_s + \frac{v_s \cdot t^*}{3}(\cos\theta_s, \sin\theta_s) \text{ and } P_2 = P_e - \frac{v_e \cdot t^*}{3}(\cos\theta_e, \sin\theta_e).$$

Then the Bézier curve between 0 and t^* controlled by P_s, P_1, P_2 and P_e is

$$B(t) = (1 - \frac{t}{t^*})^3 \cdot P_s + 3(1 - \frac{t}{t^*})^2 \cdot (\frac{t}{t^*}) \cdot P_1 + 3(1 - \frac{t}{t^*}) \cdot (\frac{t}{t^*})^2 \cdot P_2 + (\frac{t}{t^*})^3 \cdot P_e.$$

It is easy to check $||B'(0)|| = v_s$ and $||B'(t^*)|| = v_e$ as required.

Algorithm 1: Cubic Bézier Interpolation (**CubB**)

Input: P_s, P_e, θ_s, θ_e, v_s, v_e, t^* and $\langle t_1, \ldots, t_j \rangle$.
Compute P_1 and P_2.
for $i = 1 : j$ **do**
    Let $P_i = B(t_i)$;
    Let $v_i = ||B'(t_i)||$;
    Let $\theta_i = Argument(B'(t_i))$;
Output $\langle (P_i, \theta_i, v_i) : 0 < i \leq j \rangle$.

Remark 2 *In some circumstances, loops appear in the interpolation, which may not be an ideal result. The reason for loops to occur is that the Bézier curve interpolation implies the acceleration is polynomial in time and the vehicle may take a sharp turn which cannot be modelled by a polynomial acceleration. If loops is supposed to be avoided in this case, one can assume there are sudden changes of the velocity at the two endpoints and interpolate by using a quadratic Bézier curve.*

4.2 A Priori Sampling

Given a trajectory T and a privacy parameter δ, we first compute an integer $k = \lceil \frac{1}{\delta} \rceil$. Then we partition T into groups with k positions. By sampling an integer l from $\{1, ..., k\}$ uniformly, we keep the l-th position in each group and remove all other positions. Then we interpolate positions removed by using cubic Bézier interpolation.

Theorem 1 *The mechanism **SFI** is $(0, \delta)$-differentially private.*

Proof. Let T and T' be two neighboring trajectories differing only at the w-th position. By Lemma 1, it suffice to prove the process to generate the set of unchanged positions U is $(0, \delta)$-differentially private.

From the mechanism, we can see that $U = U'$ if the w-th position is removed, which is equivalent to $w \not\equiv l (mod\ k)$. Since l is uniformly sampled from $\{1, ..., k\}$, $Pr[w \equiv l (mod\ k)] = \frac{1}{k} \leq \delta$. Hence $Pr[U \neq U'] \leq \delta$. Therefore, the process to generate the set of unchanged positions U is $(0, \delta)$-differentially private. ∎

In fact, the behaviour of the privacy parameter δ is dependent on the number of positions in the underlying trajectory. To achieve the $(0, 0)$-differential privacy where $\delta = 0$, no intermediate positions can be sampled, and the output is based on the interpolation of the two endpoints only. The next strong $(0, \delta)$-differential

Algorithm 2: Remove and interpolate (**SFI**)

Input: $T = \langle (P_0, \theta_0, v_0, t_0), \ldots, (P_{n+1}, \theta_{n+1}, v_{n+1}, t_{n+1}) \rangle$ and δ.

Let $k = \lceil \frac{1}{\delta} \rceil$, $(\tilde{x}_0, \tilde{y}_0) = (x_0, y_0)$ and $(\tilde{x}_{n+1}, \tilde{y}_{n+1}) = (x_{n+1}, y_{n+1})$.

Sample an integer l from $\{1, ..., k\}$ uniformly.

for $i = 1 : n$ **do**

    if $i \equiv l (mod\ k)$, then $(\widetilde{P}_i, \tilde{\theta}_i, \tilde{v}_i) = (P_i, \theta_i, v_i)$;

    else $(\widetilde{P}_i, \tilde{\theta}_i, \tilde{v}_i) = (-)$;

Collect all the unchanged positions $U = \langle \widetilde{P}_{m_0}, \ldots, \widetilde{P}_{m_r} \rangle$.

for $i = 1 : r$ **do**

    Interpolate the removed positions between $\widetilde{P}_{m_{i-1}}$ and $\widetilde{P}_{m_i}$ and store them in $\widetilde{T}$ by **Alg** 1;

Output $\widetilde{T} = \langle (\widetilde{P}_0, \tilde{\theta}_0, \tilde{v}_0, t_0), \ldots, (\widetilde{P}_{n+1}, \tilde{\theta}_{n+1}, \tilde{v}_{n+1}, t_{n+1}) \rangle$.

privacy happens for $\delta = \frac{1}{n}$, where only one intermediate position is sampled. In other words, if the input δ is between 0 and $\frac{1}{n}$, then the mechanism is the same as that for $\delta = 0$. Generally, a non-trivial δ is one element of the discrete set $\{\frac{k}{n} : k = 0, \ldots, n\}$.

4.3 A Posteriori Sampling

In the **SFI** mechanism for small δ, few intermediate positions are sampled for the interpolation. In other words, much information between two consecutive waypoints is lost. Hence, the interpolation may not reflect the real trajectory very well. An alternative way to avoid this is to do interpolation first and then sample a sub-trajectory. Since all information is kept in the interpolation stage, the sampled sub-trajectory will be more similar to the real one.

Let $T = \langle (P_i, \theta_i, v_i, t_i)_{i=0,\ldots,n+1} \rangle$ be an input trajectory. The first step is to interpolate the curve in each time interval by using the cubic Bézier interpolation. Let $B(t)$ be the resulted Bézier-spline from $t = t_0$ to $t = t_{n+1}$. Then we sample m timestamps uniformly from these n many time intervals, say $t'_1, \ldots, t'_m$. So the intermediate trajectory T_{mid} is $\langle (B(t'_i), B'(t'_i))_{i=1,\ldots,m} \rangle$ with two endpoints, where $B'(t'_i)$ represents the direction and velocity at t'_i.

So far, T_{mid} is an alternative version of T. It can be proved the process to output T_{mid} is $(0, \delta)$-differentially private by setting $m \leq \dfrac{\ln(1-\delta)}{\ln(1-\frac{2}{n})}$. However, T_{mid} and T may not have the same timestamps. To do the comparison, we have to interpolate the positions at the timestamps of T.

Theorem 2 *The mechanism **IFS** is $(0, \delta)$-differentially private.*

Proof. Let T and T' be two neighboring trajectories differing only at the w-th position. By Lemma 1, it is sufficient to prove the process to generate T_{mid} is $(0, \delta)$-differentially private.

Algorithm 3: Remove and interpolate (**IFS**)

1 **Input:** $T = \langle (P_0, \theta_0, v_0, t_0), \ldots, (P_{n+1}, \theta_{n+1}, v_{n+1}, t_{n+1}) \rangle$ and δ.
2 Let $m = \lfloor \frac{\ln(1-\delta)}{\ln(1-\frac{2}{n})} \rfloor$.
3 Interpolate $B(t)$ from $t = t_0$ to $t = t_{n+1}$ by **Alg** 1.
4 **for** $i = 1 : m$ **do**
5 Sample a time interval index (TI_i) uniformly from $\{1, \ldots, n\}$;
6 **for** $i = 1 : m$ **do**
7 Sample a timestamp t'_i uniformly from the TI_i-th time interval;
8 Compute T_{mid} from $B(t)$ and the timestamp sequence $\langle t_0, t'_1, \ldots, t'_m, t_{n+1} \rangle$.
9 Interpolate $\widetilde{B}(t)$ from $t = t_0$ to $t = t_{n+1}$ according to T_{mid} by **Alg** 1.
10 Compute $\widetilde{T}$ from $\widetilde{B}(t)$ and the timestamp sequence $\langle t_0, t_1, \ldots, t_{n+1} \rangle$.
11 **Output** $\widetilde{T} = \langle (\tilde{P}_0, \tilde{\theta}_0, \tilde{v}_0, t_0), \ldots, (\tilde{P}_{n+1}, \tilde{\theta}_{n+1}, \tilde{v}_{n+1}, t_{n+1}) \rangle$.

From the mechanism, we can see that $T_{mid} \neq T'_{mid}$ only if some sampled timestamp falls in the w-th or the $(w+1)$-th time interval. The probability for this to happen is

$$Pr[\exists i (TI_i = w) \vee (TI_i = w+1)] = 1 - (1 - \frac{2}{n})^m \leq \delta.$$

Hence the mechanism is $(0, \delta)$-differentially private. ∎

Generally, a non-trivial δ is one element of the discrete set $\{1 - (1 - \frac{2}{n})^m : m \in \mathbb{N}\}$.

Remark 3 *In fact,* ***Line 11*** *and* ***Line 12*** *are not necessary in* ***Algorithm*** *3. The only reason for adding these two operations is to compute the MAX distance from the input trajectory, which requires the output trajectory has the same timestamp sequence as the input does. It is obvious that these additional operations would reduce the utility. Hence, if the utility measure does not require the same timestamp sequence between input and output such as DTW distance, then* ***Line 11*** *and* ***Line 12*** *can be removed.*

5 Experimental Results

In order to compare our two algorithms, we use real trajectories of ships captured in the Singapore Straits during one hour (2012-09-09 from 08:00 to 09:00 UTC time). Because of space constraint, we have selected to present results obtained from two representative trajectories with different shapes, one from a tug boat (Ship 1) and one from a cargo ship (Ship 2). As a summary, we present the average error for these two mechanisms on all real trajectories we have. For each trajectory we apply a time filter; we keep the first point and then each successive

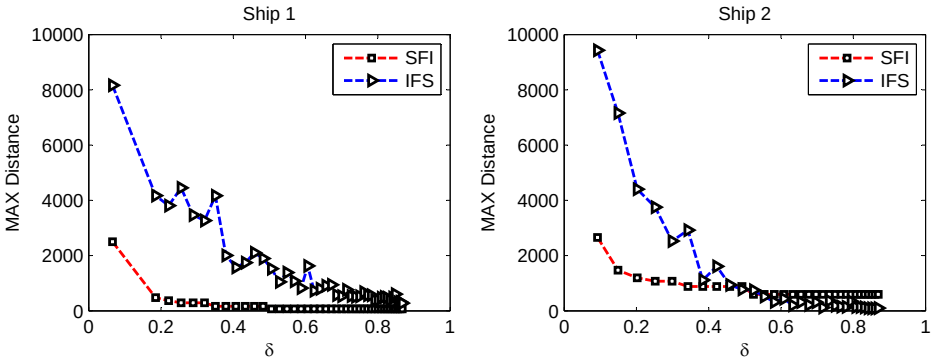

Fig. 1. MAX

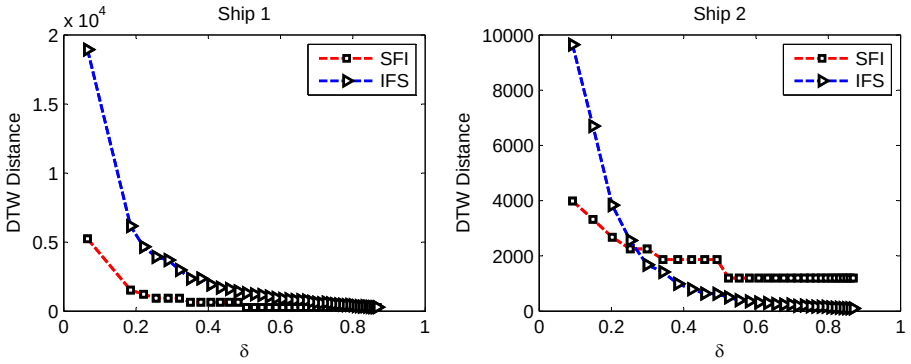

Fig. 2. DTW

point must be at least 30 seconds apart from the previous one. We then apply both algorithms with different values of δ.

In the IFS mechanism, the value of δ is determined by the number of points n of the original trajectory and by a number m of samples we want to randomly generate. We generate the values of δ corresponding to m from 1 to n. Since the SFI mechanism has no such restriction, we can then compare both mechanisms with the exact same δ value.

The SFI mechanism can be analyzed completely, as for a given δ we can have only k different outputs since we sample a natural number between 1 and k. On the other hand, the IFS mechanism cannot be analyzed for each case since we sample real numbers, so for each value of δ we generate 100 trajectories and choose to present one randomly. We then compute for both mechanisms the average distance between the original trajectory and the published trajectories, according to the *MAX* and *DTW* distances.

For the two selected representative ships, Figure 1 reports the average *MAX* error of the **SFI** and **IFS** mechanisms, and Figure 2 reports the average *DTW* error of the **SFI** and **IFS** mechanisms. Figure 3 reports the average errors for two mechanisms on all real trajectories we have, where each mechanism generates 100 trajectories for each real trajectory.

All these figures show a similar trend. The utility of **SFI** mechanism is better when δ is small. This is mainly because there are two interpolation stages in the **IFS** mechanism, where the first one generates a curve very close to the real trajectory, and the second outputs a trajectory with given timestamps. Smaller value of δ implies worse accuracy of the second interpolation. When δ reaches some value, the utility of **IFS** mechanism becomes no worse than that of **SFI** mechanism. This is because the number of sampled waypoints m goes to infinity while δ increases to 1, which will provide more accurate information. Hence, the **IFS** mechanism would be chosen for high toleration of privacy breach.

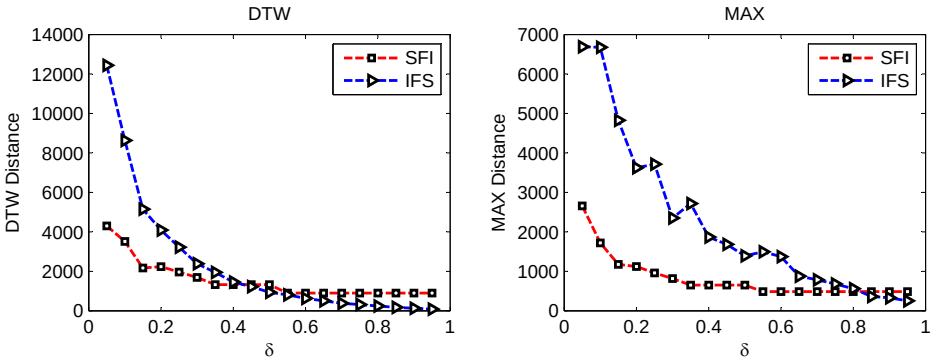

Fig. 3. Average error

Another consequence of Figure 1 and Figure 2 is that the **SFI** mechanism works better for the trajectory of ship 1 with almost all δ. This behavior is common in our experiments and the trajectory of ship 1 is representative. Hence, it is reasonable to conclude that the utility of **SFI** for not-so-smooth trajectories is better than that of **IFS** mechanism.

We can now illustrate the end result with the trajectories of two selected ships in Figure 4 and Figure 5. They illustrate the original trajectories together with their published trajectories with **IFS** and their published trajectories with **SFI**, respectively. We observe on these examples that both methods generate trajectories similar to the original trajectory and that **SFI** generates trajectories that are smoother and closer to the original one. This is exacerbated when the original trajectory is less smooth as in Figure 5.

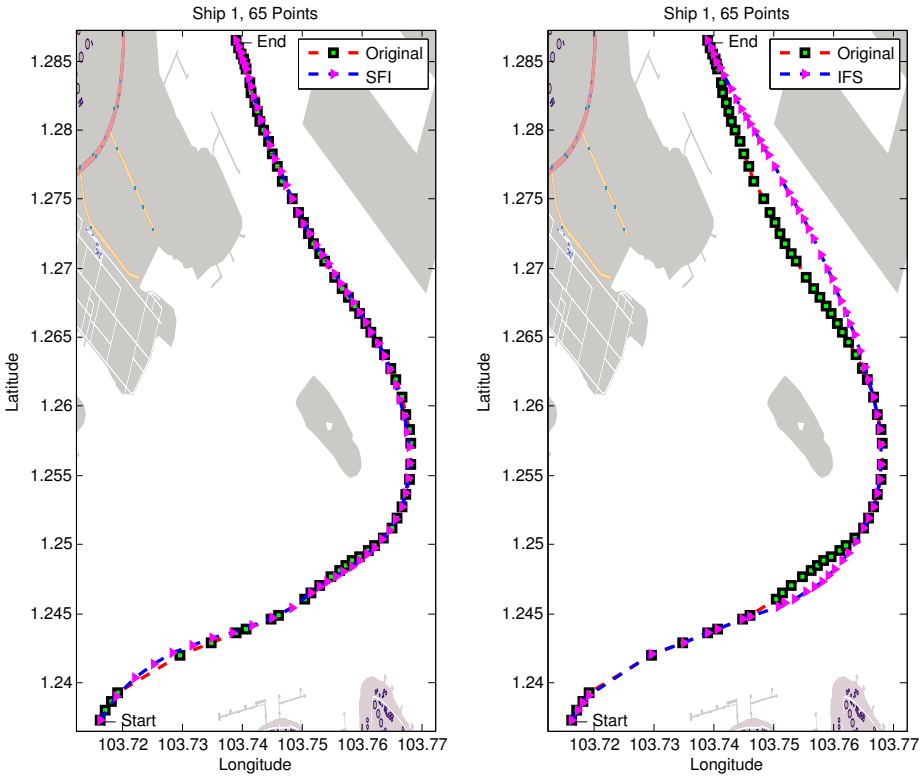

Fig. 4. Original trajectory of Ship 1 with results published by two mechanisms ($\delta = 0.1$)

6 Conclusion

The publication of the accurate trajectory of a ship is a potential menace to privacy that may threaten the security or engage the liability of the ship and its stakeholders.

We proposed two mechanisms for the publication of ship trajectories with differential privacy guarantees. The two mechanisms use a combination of sampling and interpolation to create a perturbation. The first mechanism, **SFI**, follows an a priori approach in which a trajectory is sampled and interpolated. The second mechanism, **IFS**, follows an a posteriori approach in which a trajectory is interpolated, sampled (and possibly interpolated and sampled again).

We showed that both **SFI** and **IFS** are (ϵ, δ)-differentially private with $\epsilon = 0$. We analytically and empirically compared the two mechanisms and showed that both of them are effective in publishing realistic trajectories similar to the original trajectory. We showed that the utility of **SFI** is better than that of **IFS** for smaller values of δ and not-so-smooth trajectories.

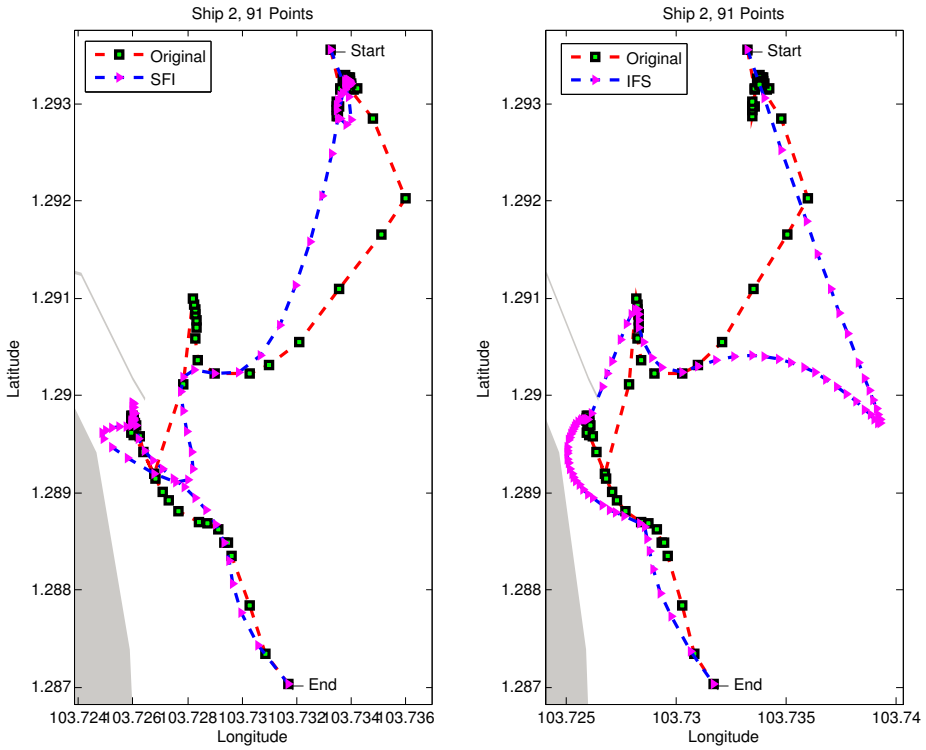

Fig. 5. Original trajectory of Ship 2 with results published by two mechanisms ($\delta = 0.1$)

We are currently fine tuning the general approaches discussed in this paper to take care of special cases such as the one discussed in Remark 2. We are also studying the extension of our techniques to take into account prescribed constraints such as further speed, acceleration and other maneuvering limits and forbidden areas.

7 Acknowledgements

This research was funded by the A*Star SERC project "Hippocratic Data Stream Cloud for Secure, Privacy-preserving Data Analytics Services" 102 158 0037, NUS Ref: R-702-000-005-305.

References

1. O. Abul, F. Bonchi, and M. Nanni. Never walk alone: Uncertainty for anonymity in moving objects databases. In *Proceedings of the 2008 IEEE 24th International*

Conference on Data Engineering, ICDE '08, pages 376–385, Washington, DC, USA, 2008. IEEE Computer Society.

2. B. Agard, C. Morency, and M. Trpanier. Mining public transport user behaviour from smart card data. In *In: The 12th IFAC Symposium on Information Control Problems in Manufacturing (INCOM*, 2006.
3. K. Chaudhuri and N. Mishra. When random sampling preserves privacy. *Advances in Cryptology-CRYPTO 2006*, pages 198–213, 2006.
4. R. Chen, B. C. M. Fung, and B. C. Desai. Differentially private trajectory data publication. *CoRR*, abs/1112.2020, 2011.
5. C. Dwork. Differential privacy. *International Colloquium on Automata, Languages and Programming - ICALP*, pages 1–12, 2006.
6. C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. *Advances in Cryptology-EUROCRYPT 2006*, pages 486–503, 2006.
7. C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. *Theory of Cryptography*, pages 265–284, 2006.
8. C. Dwork, G. Rothblum, and S. Vadhan. Boosting and differential privacy. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 51–60. IEEE, 2010.
9. J. Gehrke, M. Hay, E. Lui, and R. Pass. Crowd-blending privacy. Cryptology ePrint Archive, Report 2012/456, 2012. `http://eprint.iacr.org/`.
10. S. Gulati and B. Kuipers. High performance control for graceful motion of an intelligent wheelchair. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2008.
11. S.-S. Ho. Preserving privacy for moving objects data mining. In *Intelligence and Security Informatics (ISI), 2012 IEEE International Conference on*, pages 135 – 137, june 2012.
12. http://www.marinetraffic.com/ais/.
13. M. Klonowski, M. Przykucki, T. Strumiński, and M. Sulkowska. Practical universal random sampling. *Advances in Information and Computer Security*, pages 84–100, 2010.
14. N. Li, W. H. Qardaji, and D. Su. Provably private data anonymization: Or, k-anonymity meets differential privacy. *CoRR, abs/1101.2604*, 2011.
15. C. Mandel and U. Frese. Comparison of wheelchair user interfaces for the paralysed: Head-joystick vs. verbal path selection from an offered route-set. In *Proceedings of the 3rd European Conference on Mobile Robots (ECMR 2007)*, 2007.
16. F. McSherry and K. Talwar. Mechanism design via differential privacy. In *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*, pages 94–103. IEEE, 2007.
17. K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 75–84. ACM, 2007.
18. A. Sahraei, M. Manzuri, M. Razvan, M. Tajfard, and S. Khoshbakht. Real-time trajectory generation for mobile robots. In R. Basili and M. Pazienza, editors, *AI*IA 2007: Artificial Intelligence and Human-Oriented Computing*, volume 4733 of *Lecture Notes in Computer Science*, pages 459–470. Springer Berlin Heidelberg, 2007.