Differential Private Trajectory Obfuscation

Roland Assam, Marwan Hassani, and Thomas Seidl

RWTH Aachen University, Germany
{assam,hassani,seidl}@cs.rwth-aachen.de

Abstract. We propose a novel technique to ensure location privacy for mobility data using differential privacy. Privacy is guaranteed through path perturbation by injecting noise to both the space and time domain of a spatio-temporal data. In addition, we present to the best of our knowledge, the first context aware differential private algorithm. We conducted numerous experiments on real and synthetic datasets, and show that our approach produces superior privacy results when compared to state-of-the-art techniques.

Keywords: Spatial Database, LBS, Differential Privacy.

1 Introduction

The widespread usage of pervasive devices has made it easier to track humans and moving objects. This has garnered huge research interest on how to ensure location privacy when storing mobility data into Moving Object Databases (MOD) or Location Based Services (LBS). Recent revelation that an A-list company like Apple stored location information without its smart phone customers' knowledge highlights the elevated and immediate need to address mobility privacy. Gartner Research and its Research VP William Clark[1] are heralding Context Aware Computing as the future of computing. Sophisticated context aware data mining techniques without strong privacy guarantees will scare users from using location aware applications. The situation gets even grimmer, because some extremely good existing privacy solutions[2], [5] did not take into consideration the context of the location when anonymizing or obfuscation data. New location privacy research has the obligations to address and ensure privacy without losing the context of the original location. As a result of this, in this paper, we employ a privacy paradigm called differential privacy [3] and introduce a new privacy notion called context aware differential privacy. Providing privacy with a *very strong* privacy paradigm like *differential privacy* in context aware applications will have a profound impact w.r.t. user acceptance, or usage of context aware systems in areas such as mobile social networking, national security and trend analysis. To the best of our knowledge, this is the first location privacy work that provides a context aware location privacy using *differential privacy.*

[1] `http://www.gartner.com/technology/research/context-aware-computing/`

K. Zheng, M. Li, and H. Jiang (Eds.): MOBIQUITOUS 2012, LNICST 120, pp. 139–151, 2013.

1.1 Our Contributions

Although the theoretical strength of differential privacy has been highly and widely applauded, it is quite difficult and challenging to practically apply it in different domains as mentioned in [4], partly due to the problems that might be encountered during the derivation of the sensitivity of a metric space. In this paper, we address this challenge, and provide a differential private solution for location privacy using the exponential mechanism. In addition, we present a differential private context aware location privacy technique for moving objects. Here is a summary of our contributions:

- we derive the sensitivity of a trajectory metric space by introducing notions like Burst and Obfuscation Region (OBRegions).
- we propose a novel technique to achieve differential privacy for spatio-temporal trajectory data.
- we present to the best of our knowledge the first differential private context aware location privacy technique.

1.2 System Setup

The setup consists of a single user or multiple users carrying a GPS-enabled device and a central server that performs data perturbation. As a user or object moves, its current spatio-temporal location data is perturbed and sent to the MOD or LBS via a randomization mechanism located at the central server. Our system employs non-interactive data publishing. That is, the data is first perturbed and then published, so that any data miner can have a copy of the published perturbed data. Throughout this work, we utilize the term *Trace* to refer to a single spatio-temporal GPS point.

In many trajectory models [1], [14] significant or important locations are extracted from raw GPS data by considering only locations where an object stays for at least a given threshold time (usually termed "Stay Time"). Stay Time is the time interval between two successive traces of an object's trajectory. In this paper, any location where an object stays above a threshold stay time is called a stay point. Formally,

Definition 1. (Stay Point): *is the spatio-temporal data point of an object at a given location when its duration at that location is greater or equal to a Threshold Stay Time* T_{st}.

Definition 2. (Differential Private Context Aware Location (DP-CAL)): *is an obfuscated location that fulfills differential privacy and has a similar semantic location context as the true location from which it was derived from.*

Definition 3. (Problem Definition 1): *Given that an (outlier) object* $\mathcal{M}$ *sends raw spatio-temporal GPS data which consists of a sequence of stay points to an LBS or MOD through a trusted server, obfuscate the significant locations (stay points) of the GPS data using differential privacy at the trusted server.*

Definition 4. (PROBLEM DEFINITION 2): *Given the same object $\mathcal{M}$ and the assumptions used in Definition 3, determine a Differential Private Context Aware obfuscated trace.*

As a summary, this paper has two main goals. These include the use of differential privacy at a central server to ensure: 1) Non-context aware (or Random) obfuscation and 2) Context aware location obfuscation of the stay points of outliers or multiple moving objects.

Paper Organization. The rest of this paper is organized as follows. Section 1.3 focuses on relevant related works. In Section 2, some basic concepts of differential privacy and location obfuscation are explained. Section 3 discusses data preprocessing for differential privacy. In Section 4, we present our differential private techniques to ensure non-context and context aware differential privacy.

1.3 Related Works

Trajectory Anonymization and Location Privacy. Techniques such as [5], [6] use the spatial k-Anonymity paradigm. The topography of this paradigm typically comprises of users who send their request through a trusted server to the LBS. Anonymization is accomplished in the trusted server. This is done by selecting an area called cloaking region (CR) and for a given object's request, it ensures that at least *k-1* other object requests in that CR are sent to the LBS. Our approach is similar to these techniques only from the setup point of view. k-Anonymity is achieved in [7] by suppression, which depends on the probability of an adversary to correctly determine a trajectory sequence. [2] used inherent GPS error to propose a (k, δ)-Anonymity algorithm called Never Walk Alone (NWA) where δ represents the error radius. Our technique differs from [2] since we utilize the differential privacy paradigm while [2] is based on k-Anonymity.

Differential Privacy. Fundamental theories of differential privacy are provided in [3], [8]. We also employ some important guidelines and theories from [9] to derive a sensitivity function for the trajectory metric space which is pivotal during the derivation of noise. The data access interface of PINQ [10] and [4] are used for interactive data publishing, while ours is geared towards non-interactive publishing.

2 Background

2.1 Basics of Differential Privacy

Differential privacy is a privacy paradigm proposed by Dwork [3] that ensures privacy through data perturbation. It is based on the core principle that for any two datasets that differ in only one entry, the ratio of the probability of their outputs generated by a randomized mechanism is bounded. Specifically, this is formally given as follows.

Definition 5. (ϵ-DIFFERENTIAL PRIVACY [9]): *A randomization mechanism $\mathcal{A}$ (x) provides ϵ-differential privacy if for any two datasets $\mathcal{D}_1$ and $\mathcal{D}_2$ that differ on at most one element, and all output $\mathcal{S} \subseteq$ Range($\mathcal{A}$),*

$$Pr[\mathcal{A}(\mathcal{D}_1) \in S] \leq \exp(\epsilon) * Pr[\mathcal{A}(\mathcal{D}_2) \in \mathcal{S}]$$

where ϵ is the privacy parameter called privacy budget or privacy level.

Sensitivity. Sensitivity is defined as the maximum change that occurs, if one record is added or removed from a dataset.

Definition 6. ($\mathcal{L}_1$ SENSITIVITY [9]): *The $\mathcal{L}_1$ sensitivity of a function $f : D^n \rightarrow \mathbb{R}^d$ is the smallest number S(f) such that for all x and x' which differ in a single entry,*

$$\|f(x) - f(x')\| \leq S(f)$$

Exponential Mechanism. Differential privacy is achieved by adding noise to data. This study uses the Exponential Mechanism[8] to add noise to data. The exponential mechanism guarantees differential privacy by approximating the true value of a data with the help of a quality or utility function . Specifically, it takes in several input data and maps them to some outputs. It then uses the utility function to assign scores to all the mappings. The output whose mapping has the best score is chosen and sampled with a given probability such that differential privacy is guaranteed. This is formally given as follows.

Theorem 1. *[8] For a given input $\mathcal{X}$ and a function $u : (\mathcal{X} \times y) \rightarrow \mathbb{R}$, an algorithm that chooses an output y with a probability $\propto \exp(-\epsilon\frac{u(\mathcal{X},y)}{2\Delta u})$ is ϵ-differential private.*

Composition. [10] mentioned that there are basically two types of compositions. These include, *Sequential Composition* and *Parallel Composition.* Sequential composition is exhibited when a sequence of computations provides differential privacy in isolation. The final privacy guarantee is said to be the sum of each ϵ-differential privacy.

2.2 Location Obfuscation

Location obfuscation can be achieved by 1)Hiding Locations 2)Inserting Dummy Regions 3)Merging Regions 4) *Perturbation.* In this work, location obfuscation is accomplished by *perturbation* (using differential privacy). Location obfuscation techniques generally ensure privacy by degrading the true geographic location of an object. Most techniques [11], [12] usually define beforehand a region where the degraded location can lie on. This is then followed by the distortion of the true geographic location to any position inside the latter defined region.

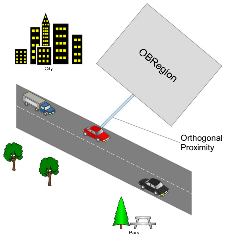


Fig. 1. An OBRegion hosts several candidate traces

3 Pre-processing Data for Privacy

3.1 Burst

The movement of an object is characterized by a sequence of stay points and points that are not stay points. Since stay points depict interesting and important locations, they are naturally the type of locations that are targeted by an adversary. Hence, they need to be properly protected. In this work, we utilize the notion of stay points to portray the mobility of an object as differential privacy problem as follows. A sequence of raw GPS data is partitioned into different data slots. Each data slot is called a Burst. Each Burst consists of one or more stay points and a trip entails one or more Bursts.

Definition 7. (Burst): *is a data slot that comprises of a finite amount of stay points.*

In terms of differential privacy, the goal is that, for any given Burst which consists of a sequence of stay points, the removal or addition of a stay point within a Burst should not reveal any information about another stay point in the Burst.

Notations. Let $\mathbf{T}$ be a trajectory or a trip consisting of a set of GPS points. The GPS points in $\mathbf{T}$ are partitioned into several Bursts $\mathbf{B}_j$ where $j \in \{1, 2, 3, ...m\}$. Let $\mathbf{p}_j$ denotes a stay point in $\mathbf{B}_j$ and each stay point $\mathbf{p}_j$ given by $(x_j,\, y_j,\, t_j)$ corresponds to a geographic position $(x_j,\, y_j)$ at time t_j.

3.2 Trajectory Obfuscation

In Section 2.2, we mentioned that in existing location obfuscation techniques, the region on which the perturbed location can fall must be defined before hand. In this work, such a region is termed the *Obfuscation Region.*

Obfuscation Region. [12] used circles to determine obfuscation regions. Our OBfuscation Region (OBRegion) is a square grid depicted in Figure 1 whose grid radius is denoted by r_o. It is connected to an arm that spans from the latter grid to the moving object. Moreover, the perpendicular distance between a moving object and the obfuscation region is called the *Orthogonal Proximity* (ρ). Using such a structure ensures higher coverage for small grid radius. The trusted server is responsible for the determination of obfuscation regions and the obfuscation of traces as follows. Once a trace arrives at the trusted server, it uses some user specified distance parameters (r_o and ρ) to determine an obfuscation region. Then, it populates this region with a finite number of *candidate obfuscation traces.* Each of these candidate traces could be chosen to replace the *stay point* of the object, thereby ensuring trace obfuscation.

Candidate Trace Generation. There are two ways by which the server can generate candidate traces. 1) By randomly picking a finite number of locations within the obfuscation region (non-context aware location obfuscation). 2) By choosing only locations within the obfuscation region that have the same location context as the true location of the object. Formally,

Definition 8. (Obfuscation Region (OBRegion)): *is a square grid region that is determined by the trusted server with the use of a user defined radius. It is also home to the* ***k*** *candidate traces generated by the trusted server. The radius of the obfuscation region is denoted by* r_o.

4 Trajectory Differential Privacy

4.1 Linking Differential Privacy to Trajectory

As aforementioned in Section 2, the exponential mechanism requires at its input among others 1) *input dataset* 2) *output range* and 3) *utility function.*

Input Dataset. The dataset of a Burst is used as the exponential mechanism's dataset. For example, assume that the dataset $\mathcal{T}_1$ corresponds to a collection of stay points within a given Burst. Adding or removing one stay point from that Burst forms a new dataset $\mathcal{T}_2$ such that $\mathcal{T}_1$ and $\mathcal{T}_2$ differ in just one single entry. $\mathcal{T}_1$ and $\mathcal{T}_2$ are sent as input dataset to a randomized mechanism $\mathcal{A}(x)$.

Output Range. Like other location obfuscation privacy techniques [11], [12], an obfuscation region that comprises of a set of locations is defined beforehand as described in Section 3.2. In addition, we indicated that the trusted server determines an OBRegion and populates it with k finite candidate obfuscation traces. An obfuscated trace is destined to fall on an OBRegion. Since the theoretical concept dictates that an output range has to be made up of a finite set of elements, we partition the OBRegion into sub regions.

The subdivision of the OBRegion is performed by the central server as follows. The grid square of an OBRegion is (vertically) divided into N equidistance sub-regions. Each of the sub-region is called *Sub-Obfuscation Region* (*Sub-OBRegion*) and it is denoted by S_i. Intuitively, after this division, the candidate traces are distributed into the N Sub-OBRegions as illustrated in Figure 2. In this paper, the output range of the exponential mechanism is given by the finite set of Sub-OBRegions that contain candidate traces. In order to prevent that no element in the output range should have a zero probability of being chosen, we have to ensure that no Sub-OBRegion is empty. Hence, Sub-OBRegions which do not contain candidate traces are discarded and the size of N is reduced. For

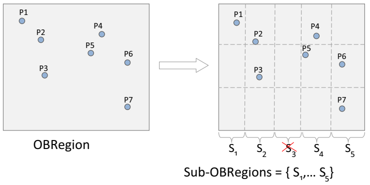


Fig. 2. Sub-OBRegion formed from an OBRegion

example, the output range R for the OBRegion in Figure 2 is given by $R = \{S_1, S_2, S_3, S_4, S_5\}$. The k candidate traces are distributed within the different Sub-OBRegions. The Sub-OBRegion S_3 is discarded since no trace is found in it and N is updated to 4.

Quality or Utility Function. As previously alluded, we intend to make the stay points private. Intuitively, the closeness between a stay point and a given Sub-OBRegion of the output range can be used to measure the quality of an obfuscated trace. Hence, the utility function for our trajectory metric space is the Euclidean distance between the stay point and the *center* of a Sub-OBRegion as formalized in Equation 1.

$$U = -dist(p_j - S_i^c) \tag{1}$$

where $dist$ denotes the Euclidean distance, p_j the stay point and S_i^c is the center of the i^{th} Sub-OBRegion. Each Sub-OBRegion is given a score based on its distance from the stay point by using this utility function. The goal is to obtain a Sub-OBRegion that is closest to the stay point. Hence, the smaller the distance, the higher the score.

Exponential Mechanism. The exponential mechanism will now map each stay point in a Burst (input dataset) to a given Sub-OBRegion (output range) and use the defined utility function to choose the optimum location which it can output as a good approximation of the original stay point. We should note that the input dataset variables (stay points from a GPS Device) are *independent* from the output range variables (traces generated by Server) since the former is retrieved from GPS readers while the latter is generated by the trusted server without any knowledge of the former.

4.2 Sensitivity Function

The utility function $U(\overline{\mathbf{p}}_j, f(\mathcal{T}_1))$ reflects how good the output obfuscated trace $\overline{\mathbf{p}}_j \in S_i$ is w.r.t. a stay point which belongs to a dataset $\mathcal{T}_1$ at a given Burst. The sensitivity of the utility function measures the maximum possible change that will occur in a trajectory metric space when one GPS stay point is added or removed from the dataset $\mathcal{T}_1$ to form a dataset $\mathcal{T}_2$ within a Burst. This sensitivity is given by :

$$S(f) = \max_{\overline{\mathbf{p}}_j \in S_i, \mathcal{T}_1, \mathcal{T}_2} |U\left(\overline{\mathbf{p}}_j, f(\mathcal{T}_1)\right) - U\left(\overline{\mathbf{p}}_j, f(\mathcal{T}_2)\right)| \tag{2}$$

Since we are dealing with geographical positions, the bounds of the sensitivity function is finite, and are defined.

4.3 Differential Private Trajectory Algorithm

Noise Addition. Algorithm 1 shows the differential private obfuscation algorithm, including the input parameters. Stay points are separated into Bursts (Line 1). Besides, the server determines the output range of the exponential mechanism by populating and computing the Sub-OBRegions with candidate traces in Line 2. Non-context aware or context aware candidate traces (Section 4.4) can be generated in Line 2 depending on the candidate trace type C_t. Non-context aware candidate traces are generated by default. The utility function in Line 3 computes the score of each Sub-OBRegion, and all candidate traces within a given Sub-OBRegion are assigned *the same score*.

The most profound step of our algorithm (Line 5) is the selection of a Sub-OBRegion based on the scores from the utility function. The exponential mechanism chooses the Sub-OBRegion using the best score with a probability proportional to $\exp\left(\frac{\epsilon}{2S(f)}.\mathbf{U}\left(\overline{\mathbf{p}}_j, f(\mathcal{T})\right)\right)$. Thus, the likelihood for a Sub-OBRegion with a better score to be selected is of an exponential magnitude. Finally, a trace within the chosen Sub-OBRegion is sampled and sent to the MOD or LBS as a differential private obfuscated trace in Line 6.

Analysis of Privacy Guarantee. Differential privacy is guaranteed for all obfuscated traces emanating from the trusted server.

Theorem 2. *Algorithm 1 is ϵ-differential private.*

Proof. In Line 5 of algorithm 1, the probability of the exponential mechanism to choose a Sub-OBRegion is given by

$$\frac{\exp\left(\frac{\epsilon}{2S(f)}.\mathbf{U}\left(\overline{\mathbf{p}}_j, f(\mathcal{T}_1)\right)\right).|S_i|}{\sum_i \exp\left(\frac{\epsilon}{2S(f)}.\mathbf{U}\left(\overline{\mathbf{p}}_j, f(\mathcal{T}_2)\right)\right) d\overline{\mathbf{p}}_j.|S_i|}$$

where $|S_i|$ is the number of Sub-OBRegions. When the best Sub-OBRegion has been chosen, a trace within the selected Sub-OBRegion is uniformly sampled with a probability.

$\propto \exp\left(\frac{\epsilon}{2S(f)}.\mathbf{U}\left(\overline{\mathbf{p}}_j, f(\mathcal{T})\right)\right)$. Since obfuscation occurs within a Burst, we utilize the longitudes, latitudes and time values of points in the Burst to extract prior knowledge about the lower and upper bounds of the sensitivity function, this means integrating $\exp\left(\frac{\epsilon}{2S(f)}.\mathbf{U}\left(\overline{\mathbf{p}}_j, f(\mathcal{T})\right)\right)$ delivers finite values. Hence sampling is being performed such that:
$Pr\left[\mathcal{A}(\mathcal{T}_1) = \overline{\mathbf{p}}_j\right] =$

$$\frac{\exp\left(\frac{\epsilon}{2S(f)}.\mathbf{U}\left(\overline{\mathbf{p}}_j, f(\mathcal{T}_1)\right)\right)}{\int_{\overline{\mathbf{p}}_j \in S_i} \exp\left(\frac{\epsilon}{2S(f)}.\mathbf{U}\left(\overline{\mathbf{p}}_j, f(\mathcal{T}_2)\right)\right) d\overline{\mathbf{p}}_j}$$

Line 5 is performed only once for a given Burst. Hence according to Theorem 1, Line 5 guarantees 1 × α-differential privacy. However, because a stay point is a spatio-temporal data which contains three dimensions, namely the X-position, Y-position and the time domain, the privacy budget needs to be carefully managed to control the cost of privacy. Using the Sequential Composition [10] described in Section 2, the total cost of privacy within a Burst to obfuscate the different dimensions is $\alpha.|D|$, where $|D|$ is the number of dimensions and $2 \leq |D| \leq 3$. This means, if all domains of the original stay point are obfuscated (i.e. $|D| = 3$) then each Burst is 3α-differential private. On the other hand, if only the spatial domains of a stay point are obfuscated, then each Burst will be 2α-differential private. Thus, for a given Burst dataset and its corresponding output range, each obfuscated trace sent to the MOD or LBS after selection by the exponential mechanism is $\alpha.|D|$-differential private.

Therefore, if an overall privacy budget ϵ is provided by the data miner, for $\alpha = \frac{\epsilon}{|D|}$, Algorithm 1 is ϵ-differential private.

Algorithm 1. Differential Private Trace Obfuscation

```
Input: Dataset $\mathcal{T}_1$, privacy budget $\epsilon$, size of Burst $\boldsymbol{n}$, OBRegion Radius $r_o$, $N$,
        Orthogonal Proximity $\rho$, Candidate Trace Type $C_t$
Output: differential private obfuscated Trace
1 Partition: Partition and group $\boldsymbol{n}$ stay points into Bursts
2 Get Output Range: Use $r_o$ and $\rho$ to determine the OBRegion. Populate
  OBRegion w.r.t. $C_t$ and divide OBRegion into $N$ Sub-OBRegions
3 Utility Function: Allocate scores to each Sub-OBRegion using the stay point
  and the utility function in Equation 1
4 Sensitivity: Get the sensitivity $S(f)$ of the trajectory metric space using
  Equation 2, $\mathcal{T}_1$ and $\mathcal{T}_2$; where $\mathcal{T}_2$ is formed by adding or removing a stay point
  from $\mathcal{T}_1$ for each Burst
5 Perturbation: Select an Sub-OBRegion and then choose a candidate trace
  within the latter region by sampling with noise whose probability is
  $\propto \exp\left(\frac{\epsilon}{2S(f)}.\mathbf{U}\left(\overline{\mathbf{p}}_j, f(\mathcal{T})\right)\right)$
6 return: the sampled trace and send to MOD or LBS
```

4.4 Context Aware Location Privacy

Context Aware computing motivations were described in Section 1. The second problem definition (Definition 4) requires the guarantee of DPCAL. As a recap, the main goal of DPCAL is to add more contextual meaning to noisy differential private traces. This is achieved using the notion of **Location Privacy Context Resolution Utility** (LPCRU).

LPCRU is a utility found at the trusted server that generates a resource pool based on the user's current location. This is used in Algorithm 1 to output a differential private context aware location. The resource pool generated by LPCRU is basically a list of location coordinates mapped to some categories. These categories have similar semantic location context to the original location.

LPCRU Candidate Traces. The LPCRU is needed to nourish Algorithm 1 with context-aware locations. The latter locations are employed by Algorithm 1 at Line 2 to populate the OBRegion with context aware locations, if the candidate trace type C_t is specified as *Context Aware Location Privacy.* The context-aware candidate traces generated by the LPCRU are used in the main algorithm (Algorithm 1 at Line 3 to 6) to produce a context aware differential private trace. In real life, there are scenarios where by an object can be found in a very sparely populated area (e.g. a desert) and there is no neighboring location which has the same location category or semantic context as the object. If LPCRU returns no candidate context-aware traces, the OBRegion which it has to populate will be empty, and this will lead to zero probability of chosen elements in the output range. This violates differential privacy. To address this problem, we stress that if no location with similar context could be found, the LPCRU will return non-context aware candidate traces to prevent zero probabilities a the output range. However, this rarely occurs in urban areas.

5 Case Study and Empirical Evaluation

The implementations were done in Java. We based our evaluations on two criteria. 1) Quantifying Privacy obtained by the user. 2) Quality and Utility of the obfuscated trace to databases and data mining. In each of these criteria, we compared our technique with that of two state-of-the-art works. They include the Never Walk Alone (NWA) algorithm [2] and the Path Confusion (PPC) algorithm [13]. Throughout this section, we will refer to these previous works as NWA and PPC, respectively.

Experimental Dataset. We conducted our experiments with one synthetic dataset and two real datasets. The Brinkhoff[2] Oldenburg synthetic dataset was used. We generated 101,070 traces for 19 objects. Besides, we utilized 90,104 traces from the GeoLife [14] real dataset. In addition, the Athens Truck[3] real dataset that entails 276 GPS trajectories of 50 moving trucks in Athens and a total of 112203 location traces was used.

5.1 Quantifying User's Privacy

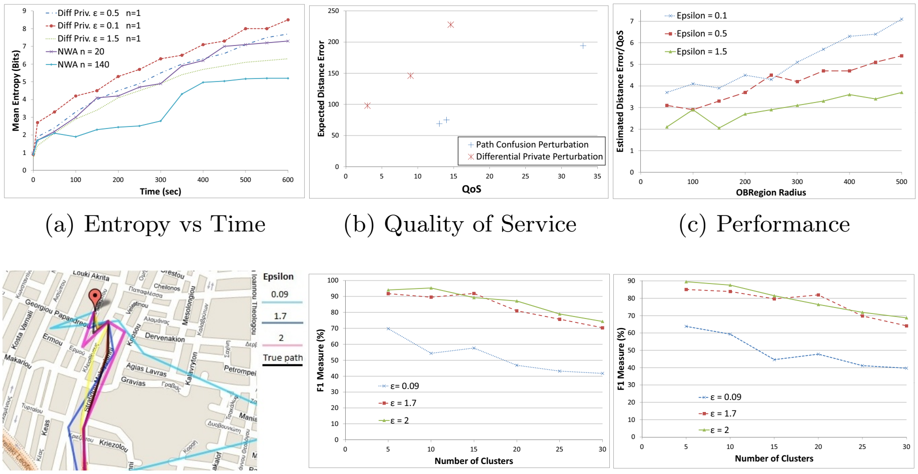
We utilized two location privacy metrics to analyze the privacy obtained by a user during obfuscation. They include 1) Expectation of Distance Error and Quality of Service (QoS) 2) Location Entropy.

Expectation of Distance Error and QoS. These privacy metrics were proposed by [13]. Expectation of Distance Error measures the accuracy by which an adversary can estimate the true position of a moving object. It is given by:

$$E[d] = \frac{1}{NT} \sum_{t=1}^{T} \sum_{i=1}^{I} p_i(t) d_i(t) \tag{3}$$

[2] http://iapg.jade-hs.de/personen/brinkhoff/generator/

[3] http://www.rtreeportal.org



(a) Entropy vs Time

(b) Quality of Service

(c) Performance

(d) LPCRU

(e) F1 Measure Context

(f) F1 Measure Random

Fig. 3. Evaluation of differential private trace obfuscation

where N is the number of objects, d_i denotes the total distance error between the true and obfuscated location, T the total observation time and $p_i(t)$ is the probability to track a user. On the other hand, **QoS** is given by:

$$QoS = \frac{1}{NT} \sum_{n=1}^{N} \sum_{t=1}^{T} \sqrt{\sum_{j=1}^{J} (\widetilde{a_n}(t) - a_n(t))^2} \tag{4}$$

where a is the domain, $a_n(t)$ is the true trace and $\widetilde{a_n}(t)$ the obfuscated trace of user n at step t.

We passed the Geolife dataset which has a GPS sampling rate of 2 to 4 seconds into the randomized mechanism. The stay point GPS data for each trajectory was partitioned into blocks of 15 stay points per Burst.We considered the movement of a user with the GeoLife dataset and perturbed the traces using our technique with an OBRegion radius of 300, 500 and 700. We computed $E[d]$ and QoS, and compared our results with that of PPC. Figure 3b illustrates these results; our technique delivers a better QoS than the PPC technique. Figure 3b orchestrates that for an OBRegion radius of 300, an adversary is expected to make an additional 18m error when comparing our method with PPC. This error distance increases as the size of the OBRegion increases.

In [13], Performance is given by the ratio of $E[d]$ to QoS. We analyzed the interplay between this ratio and ϵ for OBRegion radius $r_o \leq 500$. Figure 3c shows that for a given r_o, as ϵ decreases, the the overall performance increases. This can be explained by the fact that for each domain of the trace, smaller values of ϵ leads to greater distortion, hence causing the $E[d]$ to increase.

Entropy. Location entropy captures the uncertainty of the adversary during the inference of the correct location. Location entropy is given by:

$$H_l = -\sum P(x,y) \log_2 (P(x,y)) \tag{5}$$

where $P(x,y)$ is the probability that an object is located at position (x,y). We compared our method with the NWA technique for $\delta = 1000$. Since NWA does not anonymize the time domain, we left out the time domain of traces. We used the Geolife dataset to determine the uncertainty of the adversary for $r_o = 1000$ and $\epsilon = 0.1, 0.5, 1.5$. Figure 3a depicts the results of the experiment. Our technique produced superior entropy results when compared to the NWA, despite the fact that our technique uses just a single object while NWA uses 20 moving objects. It is important to point out that our technique insert uncertainty to each stay point of a trajectory and does not depend on neighboring objects (like in k-Anonymity). Thus, if stay points of an outlier object are passed through our randomized mechanism with low ϵ values, a very strong privacy is guaranteed.

5.2 Quality and Utility of Obfuscated Trace

LPCRU Evaluation. We conducted several experiments to evaluate the LPCRU. Figure 3d shows the map obtained when LPCRU is used for the Athens dataset with 9 stay points per Burst. The map shows distortions for ϵ values ranging from 0.09 to 2 for two categories (shops and roads). The graphical illustration shows a much larger distortion as ϵ increases. To evaluate the benefits of context aware obfuscated traces for data mining, we compared the utility of context aware and non-context aware (or random) obfuscated traces produced by Algorithm 1. We clustered each set of obfuscated trace separately using KMeans and evaluated the quality of the cluster. Figure 3e and Figure 3f shows the F1 measure results of the context aware trace and the non-context aware trace, respectively. It can be seen that the F1 Measure for context aware obfuscated trace is better.

Runtime. The time required to obfuscate traces depends on the obfuscation mode. Context aware obfuscation requires a longer time than its counterpart. For non-stream datasets, the server requires minutes to obfuscate 87K traces.

6 Conclusions

We presented a novel technique to achieve differential privacy for trajectories. Our technique extracts significant locations called stay points from raw GPS data and then obfuscates these stay points using a differential private randomized mechanism. We provide to the best of our knowledge, the first differential private context aware privacy technique and showed that our technique protects outliers.

Acknowledgments. This research was funded by the cluster of excellence on Ultra-high speed Mobile Information and Communication (UMIC) of the DFG (German Research Foundation grant EXC 89).

References

1. Ashbrook, D., Starner, T.: Using GPS to learn significant locations and predict movement across multiple users. UbiComp (2003)
2. Osman, A., Francesco, B., Mirco, N.: Never walk alone: Uncertainty for anonymity in moving objects databases. In: ICDE (2008)
3. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 1–12. Springer, Heidelberg (2006)
4. Friedman, A., Schuster, A.: Data mining with differential privacy. In: KDD (2010)
5. Kalnis, P., Ghinita, G., Mouratidis, K., Papadias, D.: Preventing Location-Based Identity Inference in Anonymous Spatial Queries. IEEE Trans. on Knowl. and Data Eng. (2007)
6. Mohamed, F.: Query processing for location services without compromising privacy. In: VLDB (2006)
7. Manolis, T., Nikos, M.: Privacy Preservation in the Publication of Trajectories. In: MDM (2008)
8. Mcsherry, F., Talwar, K.: Mechanism design via differential privacy. In: FOC (2007)
9. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265–284. Springer, Heidelberg (2006)
10. Mcsherry, F.: Privacy integrated queries. In: SIGMOD (2009)
11. Duckham, M., Kulik, L.: A Formal Model of Obfuscation and Negotiation for Location Privacy. In: Gellersen, H.-W., Want, R., Schmidt, A. (eds.) PERVASIVE 2005. LNCS, vol. 3468, pp. 152–170. Springer, Heidelberg (2005)
12. Ardagna, C.A., Cremonini, M., Damiani, E., De Capitani di Vimercati, S., Samarati, P.: Location Privacy Protection Through Obfuscation-Based Techniques. In: Barker, S., Ahn, G.-J. (eds.) Data and Applications Security 2007. LNCS, vol. 4602, pp. 47–60. Springer, Heidelberg (2007)
13. Hoh, B., Gruteser, M.: Protecting Location Privacy Through Path Confusion. In: SECURECOMM (2005)
14. Yu, Z., Li, Q., Chen, Y., Xie, X.: Understanding Mobility Based on GPS Data. In: UbiComp (2008)