LocLok: Location Cloaking with Differential Privacy via Hidden Markov Model

Yonghui Xiao #1, Li Xiong #2, Si Zhang *3, Yang Cao #4

#Department of MathCS, Emory University, Atlanta, USA
*Department of MathCS, Jianghan University, Wuhan, China
{ 1yonghui.xiao, 2lxiong, 4ycao31}@emory.edu, 3sz04eu@jhun.edu.cn

ABSTRACT

We demonstrate LocLok, a LOCation-cLOaKing system to protect the locations of a user with differential privacy. LocLok has two features: (a) it protects locations under temporal correlations described through hidden Markov model; (b) it releases the optimal noisy location with the planar isotropic mechanism (PIM), the first mechanism that achieves the lower bound of differential privacy. We show the detailed computation of LocLok with the following components: (a) how to generate the possible locations with Markov model, (b) how to perturb the location with PIM, and (c) how to make inference about the true location in Markov model. An online system with real-word dataset will be presented with the computation details.

1. INTRODUCTION

With the technology advances in smartphones with localization capabilities, location based applications have been tremendously popular in people's lives. Location-based services (LBS) [11, 5] range from searching points of interest to location-based games and location-based commerce. Location-based social networks allow users to share locations with friends, to find friends, and to provide recommendations about points of interest based on their locations.

A concern of the location based applications is location privacy [2]. Because locations contain a lot of sensitive information such as one's religion and health condition (when a user goes to church or hospital), it is considered as private information of users. However, to enable the location based applications, users have to provide their locations to the respective service providers or other parties. This location disclosure raises important privacy concerns since digital traces of users' whereabouts can expose them to attacks ranging from unwanted location based spams/scams to blackmail or even physical danger.

Challenges. Most existing location preversing solutions in the literature are based on location obfuscation which replaces the exact location with an area (location generalization) or a noisy location (location perturbation) (e.g. [8, 1]). Many of them only consider static scenarios or perturb the location at single timestamps without considering the temporal correlations of a moving user's locations, and hence are vulnerable to various inference attacks. For example, by combining the knowledge of temporal correlations, the static noisy location can be easily disclosed to adversaries. Such temporal correlations are usually described by Markov model to reflect the patterns such as user moving habits or road network constraint. Because the true location is always hidden from adversaries and service providers, it is a hidden Markov model.

Differential privacy [6] has been considered as an accepted notion for privacy preservation. Initially it was proposed to protect aggregated statistics of a dataset by limiting the knowledge gain of the neighboring databases whether a user is in a dataset or not. Applying differential privacy on location data was conducted in recent literature. In particular, several works (e.g. [4, 7, 3]) have applied differential privacy on location or trajectory data but in a data publishing or data aggregation setting. In contrast, in our setting of continual location sharing, the protection needs to be enforced on the fly for *a single user*. For example, the recent work [1] proposed a notion of geo-indistinguishability which bears some similarity to differential privacy by changing the notion of neighboring databases.

Several challenges emerge when adopting differential privacy in our setting. First, standard differential privacy only protects user-level privacy (whether a user opts in or out of a dataset); while in our setting, the protection needs to be enforced for *a single user*. Hence the user cannot opt out of the system, otherwise there is no data to protect. Second, temporal correlations have to be considered to account for the road networks or the user's moving patterns.

Contributions. In this paper, we demonstrate the location cloaking system [13] to preserve location privacy with differential privacy. Our framework is shown in Figure 1, where we have a moving user with location stream who needs to share the locations to some service providers or other parties. The user's true locations are kept to only the user. The noisy locations are released by the privacy mechanisms to the service providers, and visible to adversaries. To preserve location privacy, we tackle the temporal correlations described by Markov model, which are assumed to be public. Furthermore, we assume the release mechanism is transparent to adversaries, who, in the worst case, can even have the knowledge of all historically released locations from the

This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-nd/4.0/. For any use beyond those covered by this license, obtain permission by emailing info@vldb.org.
Proceedings of the VLDB Endowment, Vol. 10, No. 12
Copyright 2017 VLDB Endowment 2150-8097/17/08.

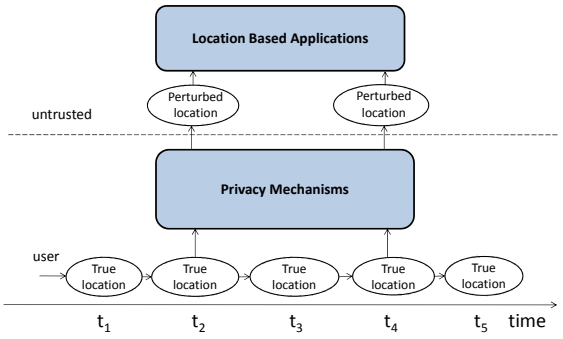


Figure 1: Problem setting

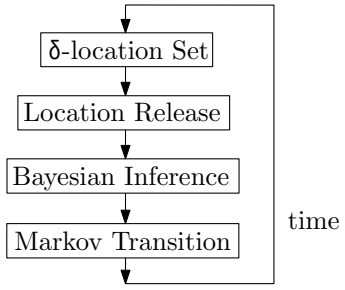


Figure 2: System Overview

user.

First, we demonstrate an extended differential privacy definition based on the notion of δ-*location set* we developed in [13]. In our problem, location changes between two consecutive timestamps are determined by temporal correlations modeled through a Markov chain [12, 9]. Accordingly we propose a "δ-location set" to include all probable locations (where the user might appear) at current timestamp. To protect the true location, we "hide" it in the δ-location set in which the elements are not distinguishable to each other.

Second, we show how to perturb the true location with an efficient differentially private mechanism, called planar isotropic mechanism (PIM), based on δ-location set. To our knowledge, PIM is the first optimal mechanism that achieves the lower bound of differential privacy. The trick is that location data is only two-dimensional (or three-dimensional at most). Hence we can transform the location to an isotropic space to conduct the perturbation so that the optimality of the error can be guaranteed.

Third, we present the continual location sharing system by combing the Markov model and the PIM together. We show that even with the temporal inference, the location at current timestamp can still be protected by the δ-location set based differential privacy.

Software Availability. Our demonstration using the GeoLife data and OpenStreetMap will be put on the website `http://www.loclok.com`. Besides our demonstration system, we also have a prototype iPhone app "LocLok" in Apple's app store. The source code of the app can also be found at `https://github.com/yhxiao/gitloc`.

2. SYSTEM OVERVIEW

We partition a map into grids where each grid is a state in Markov model. Then a user's true location is denoted by a state (grid) number in a Markov model.

At each timestamp in LocLok, three components are calculated: (1) a δ-location set is generated in Markov model to find the probably locations of the user; (2) a noisy location $\mathbf{z}$ is released using a differentially private mechanism, i.e. PIM; (3) The released location is used to make inference about the true location. Then by Markov transition, the probability after inference will be used in the next timestamp. The three components iterates as a loop over time. Figure 2 shows the system overview. The last step "Markov transition" is a standard movement with Markov transition matrix.

2.1 Generating δ-location set

The idea of differential privacy is to hide a database in its neighboring databases, which are derived by adding or removing a record in the database. Similarly, we can hide the user's true location in possible locations where the user may appear. On the other hand, hiding the true location in any impossible locations does not work because the adversary already knows the user cannot be there. We use the concept of δ-location set to denote the possible locations of the user.

To incorporate temporal correlations, we assume the Markov model is public. Hence adversaries can also use the Markov model to derive the probability of the true location of a user. At any timestamp, say t, a prior probability of the user's current location can be derived, denoted by $\mathbf{p}_t^-[i] = Pr(\mathbf{u}_t^* = \mathbf{s}_i | \mathbf{z}_{t-1}, \cdots, \mathbf{z}_1)$.

We use a parameter δ to derive the possible locations of the user. δ-location set $\Delta\mathbf{X}_t$ is a set containing minimum number of locations that have prior probability sum no less than $1-\delta$. Essentially, it is a set of the most probable locations of the user after removing the unlikely locations.

$$\Delta\mathbf{X}_t = min\{\mathbf{s}_i | \sum_{\mathbf{s}_i} \mathbf{p}_t^-[i] \geq 1-\delta\}$$

Then differential privacy can be guaranteed on $\Delta\mathbf{X}_t$ if for any two locations $\mathbf{x}_1$ and $\mathbf{x}_2$ in $\Delta\mathbf{X}_t$ $\frac{Pr(\mathcal{A}(\mathbf{x}_1)=\mathbf{z}_t)}{Pr(\mathcal{A}(\mathbf{x}_2)=\mathbf{z}_t)} \leq e^\epsilon$ holds where $\mathbf{z}_t$ is the released location from any mechanism $\mathcal{A}$.

2.2 Location Release Mechanism

Instead of adopting the standard Laplace mechanism to release the "noisy" location, we use a planar isotropic mechanism (PIM) to release the differentially private locations. The novelty is that we transform the data release mechanism (K-norm mechanism [10]) into an isotropic space, generate the "noisy" location, and transform the noisy location back to the original space. In this way, the released location achieves the lower bound of differential privacy [13].

To denote the sensitivity function of differential privacy, we use sensitivity hull to denote the geometric sensitivity of the location. Given the δ-location set, we can derive a convex hull K' covering all the points in $\Delta\mathbf{X}$ by $Conv(\Delta\mathbf{X})$ where $Conv()$ is a function of convex hull. Then the sensitivity hull K can be derived by covering all the differences among every pair of points in K'.

$$K = Conv(\Delta\mathbf{V})$$
$$\Delta\mathbf{V} = \bigcup_{\mathbf{v}_1, \mathbf{v}_2 \in \text{ vertices of } K'} (\mathbf{v}_1 - \mathbf{v}_2)$$

To implement the planar isotropic mechanism (PIM), we transform the sensitivity hull to its isotropic space. This is computationally feasible because in 2-dimensional space the transformation is in constant time. To derive the transformation matrix $\mathbf{T}$, we uniformly draw random samples in K. When the number of samples grows, the matrix $\mathbf{T} = \left(\frac{1}{l}\sum_{i=1}^{l}\mathbf{y}_i\mathbf{y}_i^T\right)^{-\frac{1}{2}}$ becomes stable where $\mathbf{y}_1, \cdots, \mathbf{y}_l$ are the random samples from K. Then we transform the sensitivity hull K to $\mathbf{T}K$.

Next we generate a random variable r from Gamma distribution $\Gamma(3, \epsilon^{-1})$. Let $\mathbf{z}' = r\mathbf{z}'$. Then we transform the point $\mathbf{z}'$ to the original space by $\mathbf{z}' = \mathbf{T}^{-1}\mathbf{z}'$. Finally, the released location is $\mathbf{z} = \mathbf{x}^* + \mathbf{z}'$.

2.3 Inference

Similar to the forward-backward algorithm in hidden Markov model, the released location $\mathbf{z}_t$ can be used to make better estimation of the true location. For privacy reasons, we assume the data release mechanism, i.e. PIM, is transparent to adversaries. Thus the probability distribution of $\mathbf{z}_t$ is known. To derive the probability of $\mathbf{s}_i$ being the true location, we first transform $\mathbf{z}$ and $\mathbf{s}_i$ back to the isotropic space $\mathbf{z}'_t = \mathbf{T}\mathbf{z}$, $\mathbf{s}'_i = \mathbf{T}\mathbf{s}_i$. Next the emission probability $Pr(\mathbf{z}_t|\mathbf{u}_t^* = \mathbf{s}_i)$ can be derived in the isotropic space. Then we use Bayesian rule to derive the posterior probability $Pr(\mathbf{u}_t^* = \mathbf{s}_i|\mathbf{z}_t)$, which is also the inference of $\mathbf{u}_t^*$ given all the released locations $\mathbf{z}_1, \cdots, \mathbf{z}_t$.

In the next timestamp $t+1$, the prior probability can be derived as $\mathbf{p}_{t+1}^- = \mathbf{p}_t\mathbf{M}$ where $\mathbf{M}$ is the transition matrix of Markov model. Then the three components repeat again as in Figure 2.

3. DEMONSTRATION

We demonstrate the functionalities and utilities of LocLok using the real-world dataset GeoLife. The computation details will also be included in the demonstration.

Dataset. Geolife data [14] was collected from 182 users in a period of over three years. It recorded a wide range of users' outdoor movements, represented by a series of tuples containing latitude, longitude and timestamp. The trajectories were updated in a high frequency, e.g. every $1 \sim 60$ seconds. We clean the data by extracting the locations every 5 minutes. Then we extracted all the trajectories within the 5th ring of Beijing to train the Markov model. Because the map granularity affects the Markov model and the corresponding trajectories, we show three layers of the same data by partitioning the map into 25×25, 50×50 and 100×100 grid.

Basic Functions. First, we choose a trajectory of a user in one day. For example, Figure 3 shows a trajectory of a user, marked in the grid coordinates. The green point indicates the true location at current time. When the user moves, the probability of the location can be derived by Markov model. Figure 4 shows the δ-location set of the user, containing all the probable locations the user might appear based on previously perturbed (released) locations and the Markov transition.

Second, we show how to release a noisy location $\mathbf{z}_t$ with PIM. Given the convex hull of δ-location set in Figure 5a, we first draw the sensitivity hull in Figure 5b by moving the convex hull around while containing the origin within the convex hull. Then the covered area forms the sensitivity hull K. Third, we show how to transform the sensitivity hull to its isotropic space. We randomly draw samples in K, and derive the transformation matrix $\mathbf{T} = \left(\frac{1}{l}\sum_{i=1}^{l}\mathbf{y}_i\mathbf{y}_i^T\right)^{-\frac{1}{2}}$ where $\mathbf{y}_1, \cdots, \mathbf{y}_l$ are the random samples from K. Thus by applying $K_I = \mathbf{T}K$, the sensitivity hull can be transformed to the isotropic space K_I, shown in Figure 5c.

Finally, we generate the noisy location $\mathbf{z}_t$. By Bayesian inference, the posterior probability of the true location $\mathbf{p}_t^+$ can be derived. Then at the next timestamp, the process repeats again. The released trajectory is the series of noisy locations over time. For example, Figure 6a is the true trajectory of a user on a map. Then the released trajectory is shown in Figure 6b. We can see that the released trajectory is still close to the true trajectory, and ϵ-differential privacy on δ-location set is preserved at each timestamp where $\epsilon = 1$ and $\delta = 0.01$.

Advanced Functions. We also demonstrate two advanced functions: (a) how the parameters impact the result, (b) how the map granularity changes the result. We will allow different parameters of ϵ and δ perform on the same dataset to compare the different results. We show that with larger ϵ, the error in the released location is smaller. Whereas with larger δ, the δ-location set becomes smaller. Although this may lead to better utility, it may also cause that true location is excluded in the δ-location set. Even with the solution of surrogate [13], we need to tune the value of δ for better utility. In general, if both ϵ and δ are small, the area of δ-location set will become larger and larger after a long time as Markov model converges. In this case, a larger ϵ is preferred for better utility.

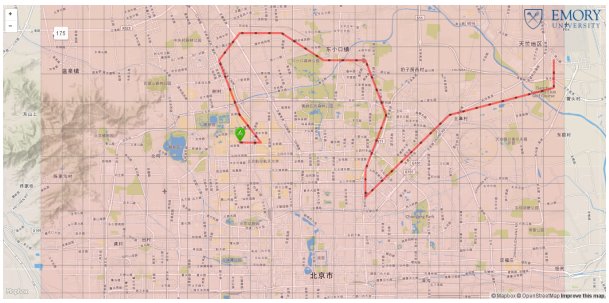


Figure 3: A trajectory in GeoLife

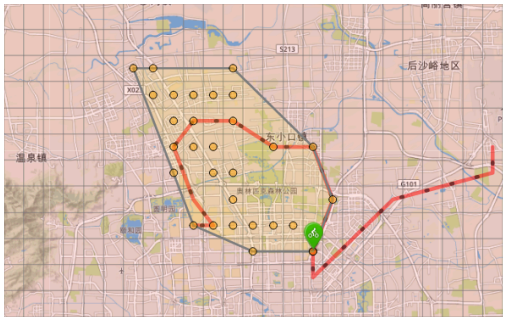
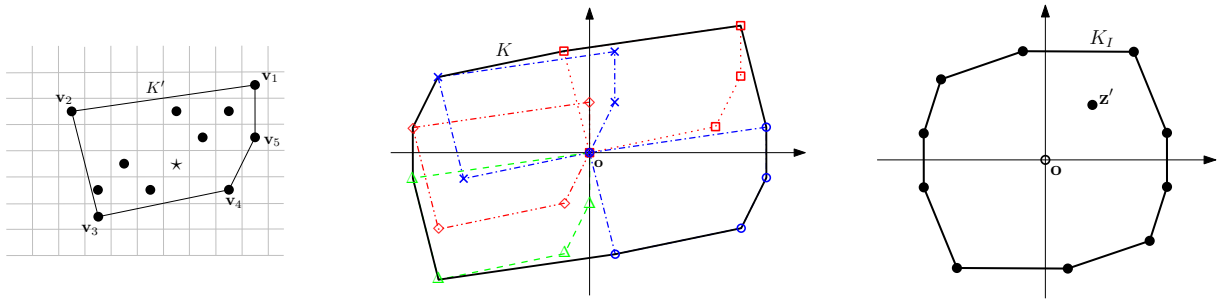
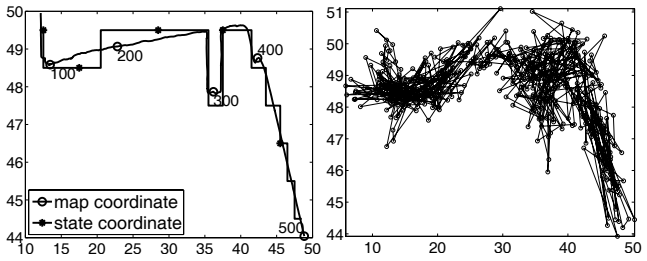


Figure 4: δ-location set of the user



(a) (b) (c)

Figure 5: (a) Convex hull of $\Delta\mathbf{X}$. (b) Finding the sensitivity hull K. (c) Transform K to isotropic position K_I. Sample a point $\mathbf{z}'$.



(a) True trace (b) PIM released trace

Figure 6: Performance over time: (a) The true (original) trajectory; (b) the released trajectory.

We also show the impact of different map granularity by showing the same data under 3 partitioning size: 25×25, 50×50 and 100×100. Intuitively, with finer granularity and larger partition size, the trajectory contains more moving details with smaller grid. Furthermore, with finer granularity, the trained Markov model is more diverging, which means the probability of staying in the same grid will be smaller. This leads to the more possible locations and complicated sensitivity hull. On the other hand, with coarse granularity, the information of the trajectory becomes less, and the diagonal values in the transition matrix becomes larger. Hence the probability of staying at the same grid rises. The moving speed of the user may become vague over time.

User Interface. The interface of our system is the map page from OpenStreetMap. We will give options to select different trajectories in the dataset. After showing the true trajectory, we let audience choose the map granularity and the parameters. Then the system will show the true location, δ-location set, the sensitivity hull, and the noisy location on the map until the moving user reaches the end of the trajectory.

4. ACKNOWLEDGEMENT

This research is supported by NSF under grant No. 1618932 and the AFOSR DDDAS program under grant FA9550-12-1-0240. We thank the NSF I-Corps program for helping the authors explore the opportunities of the real-world commercialization.

5. REFERENCES

[1] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. CCS '13, pages 901–914. ACM, 2013.

[2] A. R. Beresford and F. Stajano. Location privacy in pervasive computing. *Pervasive Computing, IEEE*, 2(1):46–55, 2003.

[3] Y. Cao, M. Yoshikawa, Y. Xiao, and L. Xiong. Quantifying differential privacy under temporal correlations. *ICDE*, pages 821–832, 2017.

[4] R. Chen, B. C. Fung, B. C. Desai, and N. M. Sossou. Differentially private transit data publication: a case study on the montreal transportation system. KDD '12, pages 213–221. ACM, 2012.

[5] A. Dey, J. Hightower, E. de Lara, and N. Davies. Location-based services. *Pervasive Computing, IEEE*, 9(1):11–12, 2010.

[6] C. Dwork. Differential privacy. In *Automata, languages and programming*, pages 1–12. Springer, 2006.

[7] L. Fan, L. Xiong, and V. S. Sunderam. Differentially private multi-dimensional time series release for traffic monitoring. In *DBSec*, pages 33–48, 2013.

[8] B. Gedik and L. Liu. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *Mobile Computing, IEEE Transactions on*, 7(1):1–18, 2008.

[9] M. Götz, S. Nath, and J. Gehrke. Maskit: Privately releasing user context streams for personalized mobile applications. SIGMOD, 2012.

[10] M. Hardt and K. Talwar. On the geometry of differential privacy. In *STOC*, pages 705–714. ACM, 2010.

[11] I. A. Junglas and R. T. Watson. Location-based services. *Communications of the ACM*, 51(3):65–69, 2008.

[12] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux. Quantifying location privacy. IEEE SP '11, pages 247–262, Washington, DC, USA, 2011.

[13] Y. Xiao and L. Xiong. Protecting locations with differential privacy under temporal correlations. CCS '15, pages 1298–1309. ACM, 2015.

[14] Y. Zheng, X. Xie, and W.-Y. Ma. Geolife: A collaborative social networking service among user, location and trajectory. *IEEE Data Eng. Bull.*, 33(2):32–39, 2010.