An Adaptive Privacy-Preserving Scheme for Location Tracking of a Mobile User

Jindan Zhu,[1] Kyu-Han Kim,[2] Prasant Mohapatra,[1] and Paul Congdon[2]

[1]Department of Computer Science,
University of California at Davis, Davis, CA 95616
{jdzhu, pmohapatra}@ucdavis.edu

[2]Hewlett-Packard Laboratories,
Palo Alto, CA 94304
{kyu-han.kim, paul.congdon}@hp.com

***Abstract*—Many popular mobile applications require the continuous monitoring and sharing of a mobile user's location. However, exploiting a user's location leads to disclosing sensitive information about the users daily activity. Several location privacy-preserving schemes have been proposed, but it remains challenging for a user to achieve visibility of the associated threats as well as to control the impact of those threats. This paper presents an adaptive location privacy-preserving system (ALPS) that allows for a user to control the level of privacy disclosure with different quality of location-based service (LBS). We have identified key attack models on location tracking using powerful map-matching algorithms, and then defined a scheme that allows a user to control the privacy of tracking information. We have implemented ALPS on Android OS and evaluated the implementation extensively via trace-based simulation, showing the effectiveness of user-controllable privacy preservation.**

I. Introduction

Location based services are evolving from requesting single location to recording a trace of continuous locations in performing tasks such as trajectory sharing, participatory sensing, and destination/intention predicting. This situation introduces another dimension of complexity and confusion for users in controlling their privacy of shared locations. Existing studies on location privacy preservation mechanisms (LPPM) mainly focus on protecting privacy of individual locations. Extending them to preserving privacy of location trace (i.e. trace privacy) is not exactly straightforward, due to the spatial and temporal correlative nature of location trace as to be explained in Section II. Furthermore, users of aforementioned new LBSs demand personalized privacy protection to their location traces, in which privacy profile can be adaptively adjusted for individual segments in a trace in order to achieve a balance between privacy and QoS within the trace. Traditional schemes find it difficult to adapt to this requirement without degrading performance or revealing user's privacy preference over the trace. On the other hand, advances in map-matching algorithm equipping adversary with contextual information impose a more challenging issue. For instance, authors in [1] propose an accurate map-matching algorithm for location tracking of a mobile user that can be potentially used by privacy adversaries to accurately reconstruct a user's actual trace, even with a highly obfuscated trace.

In facing these challenges, this paper presents an Adaptive Location-tracking Privacy-preserving System (ALPS) that allows a user to dynamically control the level of *trace privacy* disclosed to LBS or an adversary. ALPS runs on a mobile device and provides context-aware perturbation mechanisms as well as attack emulation capability for privacy preservation and potential privacy threat quantification. ALPS, in its core, takes a two-tier approach to perturbation, in which the system in the first tier injects artificial perturbations into the location trace and then, in the second tier, conforms and smoothens the perturbation to mitigate any hints that might be useful to a potential adversary. ALPS provides several control knobs for the perturbation and the conformation tier, allowing for a user to adaptively adjust the privacy levels in practical settings. Furthermore, ALPS exploits contextual information and previous location release history by integrating them into various adversary emulator, in order to estimate potential trace privacy threats as well as to provide a user feedback to adjust privacy settings.

We have evaluated ALPS through both implementation and trace-based simulation. We implemented ALPS on Android OS and collected location traces in both campus (Davis, CA) and urban areas (Mountain View, CA). In addition, we have identified and developed three map-matching adversary models, and have compared our implementation against the three models. Trace-based evaluation of ALPS shows that our two-tier approach enables a user to effectively protect location trace privacy. In addition, ALPS is able to selectively expose or conceal sensitive locations.

The rest of this paper is organized as follows. Section II motivates the need for trace privacy and describes key design goals to meet the need. Section III discusses the privacy and adversary model that we consider in this paper. Section IV presents our proposed system to protect the trace privacy of a mobile user. Section V shows evaluation results on our proposed system. Section VI discusses related work, and finally we conclude this paper in Section VII.

II. Motivation

Most existing schemes are competent at protecting location privacy as well as anonymity, but not necessarily trace privacy. Trace privacy is vulnerable due to the correlated nature of continuous location samples. For example, consider spatial cloaking [2], a popular set of LPPMs. Correlation property such as maximum velocity enables an adversary to predict user position distribution from current cloaking region (CR), so that subsequently disclosed CR can be effectively shrunk by intersecting which with the prediction. Countermeasures [3] take correlation into consideration by deferring the disclosure of CR until velocity constrains are satisfied, which is questionable for continuous LBS since deferral to a scheduled disclose itself implies a conflict, and analysis of the delay can be later used to shrink CR when disclosed. Moreover, it is difficult for spatial cloaking to meet a personalized privacy requirement. The variation in cloaking parameters directly reflects on the obfuscation output as CR size, and it may allow an adversary to easily identify user‘s privacy preference and discover sensitive CRs by interpolating immediate adjacent insensitive regions in the trace. Another promising approach is mixing the actual trace with multiple realistic dummy traces [4], [5]. However, the additional communication overhead incurred (e.g., LBS query/response) makes it formidable for practical deployment.

Challenges with spatial cloaking stem from the notion that *the actual location must be enclosed by the cloaking region*. However, this requirement is not necessarily mandatory for the majority of LBS applications. Both applications and users can tolerate the obfuscation error to a degree, as seen in various and widely used coarse-grained localization technologies (e.g., Cellular-ID look-up). Dummy traces requires a user to create completely bogus traces to achieve trace privacy, at the expense of communication overhead.

Perturbation provides an alternative solution. Perturbation obfuscates location sample by adding noise to intentionally reduce its accuracy. Compared to the spatial cloaking approach, perturbation disables the ability of an adversary to make a calculated guess about the actual trace as well as its obfuscation parameters. The additive noise provides a more natural and flexible way to destroy the correlation exhibited between consecutive samples in the actual trace. If devised carefully to withhold additional information about the characteristics of the noise, it would be a daunting task for the adversary to remove the noise and recover the actual trace. In addition, by controlling the magnitude of the additive noise on a per-sample basis, the personal privacy requirement can be easily fulfilled with little computational complexity. Since perturbation is performed directly on the original trace, no additional communication overhead is generated.

Motivated by the above observations, this paper proposes a novel LPPM based on context-aware perturbation that aims to protect the trace privacy of a specific mobile user with the following design goal in mind:

- *Linkage attack resistance*: The correlation between continuous location samples should be re-arranged, so that adversary cannot effectively undermine the performance of LPPM through a linkage attack.
- *Personalized privacy support*: Based on personalized privacy preference profile that includes (in)sensitive regions with privacy levels, the LPPM should be able to enforce regional privacy with intuitive control parameters, while preserving a user's privacy preference. Transition between regions with different privacy levels applied should be smooth enough to prevent adversary from detecting and exploiting it for gaining insight of privacy preference, yet should converge quickly to minimize QoS degradation for insensitive region.
- *Mobile-centric approach*: There exists no centralized trusted third party between a mobile user and the untrusted LBS provider. The LPPM should collocate on the mobile device and should be light-weight enough to perform online obfuscation.
- *Energy and communication overhead*: The LPPM should support various location providers, which are increasingly considering energy-aware LBS design. Unnecessary communication transmission should be minimized to avoid inflicting additional data charges on a mobile user.

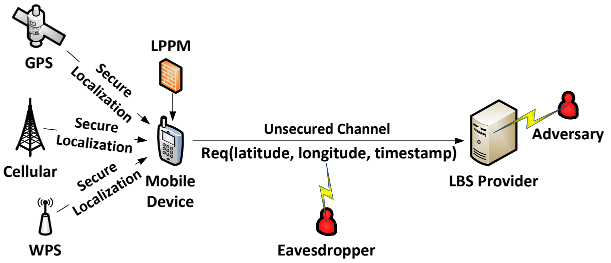


Fig. 1. The LBS Model

III. LBS, Privacy and Adversary Models

We define models of LBS, privacy and adversary that we will use throughout this paper.

A. Location-Based Service (LBS) Model

In this paper, we consider a continuous location sharing and tracking (e.g., Google Latitude [6]) as an LBS model. Figure 1 illustrates this model. There are two parties in the model: a mobile device and an untrusted LBS provider. We assume the mobile device can securely localize itself using various localization technologies, such as Global Positioning System (GPS), Cellular-ID look-up (CID), and Wi-Fi Positioning System (WPS). The mobile device will periodically send location updates, containing a timestamp and the users current geodetic coordinates to the LBS provider. However, we do not assume a secure communication channel between them, and thus an eavesdropper may intercept outgoing location updates as well as the mobile user's identity.

B. Personalized Location Trace Privacy

Trace privacy can be categorized as a special type of location privacy [7], and it primarily concerns the user's control over when, how, and to what extent the continuous location information is communicated to others. Location samples in a trace are correlated temporally as well as spatially. Therefore, a LPPM must be particularly careful to not assume independence when performing obfuscation. This prevents any untrusted party from inferring the actual trace. We find that perturbation is one obfuscation approach that is inherently preferable in this context—we will further elaborate this finding in the following section. In the rest of this paper, we use perturbation and obfuscation interchangeably.

A privacy requirement on trace privacy may vary with factors such as time, space, and trustworthiness of LBS. In this work, we mainly focus on the spatial factor, dividing space into sensitive regions and insensitive regions. Sensitive regions require higher privacy than the insensitive ones. On the other hand, insensitive regions might have stringent QoS requirement (e.g., proper LBS response with accurate location information).

User should be allowed to personalize these needs into a privacy profile, and the LPPM should adapt its parameters for location samples in accordance with the sensitive/insensitive classification.

Since the identity of the mobile user is assumed to be known by the adversary, approaches for protecting anonymity, such as *k-anonymity* and *entropy-based* are not considered. By dismissing spatial cloaking, we also rule out metrics like *l-diversity*. Instead, we consider a **distortion-based metric** a more appropriate candidate for evaluating both privacy and LBS quality in a perturbation-based obfuscation scheme (e.g., [8]). This metric assumes an adversary performs reconstruction of an observed trace, and the distortion between the reconstructed trace and the actual trace is used as privacy indicator of how good the obfuscation is. A single reconstructed trace with maximum likelihood is used in the distortion calculation. Euclidean distance is used as the distance function in the calculation of per-sample distortion, and trace distortion is defined as the average of per-sample distortion over the entire trace.

C. Adversary Model

All parties that have access to the perturbed trace, without knowing the actual trajectory, are assumed to be potential adversaries. The primary objective of an adversary is to reconstruct the mobile users actual trajectory from the user's perturbed trace. The adversary is also interested in extracting the personalized privacy preference of the mobile user, by exploiting the density and turbulence in the perturbed trace.

The reconstruction should be automated and, therefore, manual inspection is out of our scope. In this work, we also study only a passive adversary that obtains a user's trajectory information after obfuscation, as opposed to an adversary that actively collects trajectory information through means such as stalking or engineered encounters. In addition to user identity and location updates, we assume an adversary can also acquire the following items (or side information) to assist in the reconstruction process:

- The adversary is assumed to know about the localization technologies available on the mobile device, and the accuracy and granularity estimation of the respective technology for the interested region.
- The obfuscation algorithm in use is known to the adversary, but not the specific parameters for individual run of obfuscation.
- The adversary might have an estimate of the general mobility pattern in the interested region. Parameters such as average speed, maximum speed, or time of travel can be estimated through posted speed limits or obtained from publicly accessible sources.
- The adversary could have the knowledge of geographical topology. A variety of maps of the interested region are easily obtained from various sources.

We also assume that the adversary will focus on the macroscopic trajectory (e.g., a collection of paths that a user has travelled). Specifically, if a user dwelled indoor for an arbitrary period of time, the adversary is more interested in identifying the entrance and exit trajectory, rather than movements inside the facility.

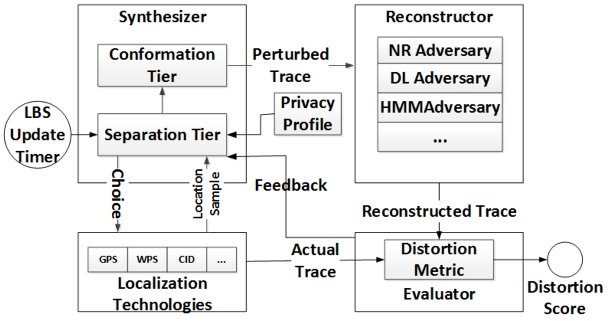


Fig. 2. System Architecture of ALPS

An adversary who fits this model and possesses the aforementioned side information is denoted as a *Map-Matching Adversary*, since their task is similar to that of a map matching algorithm that deals with location samples with large error and outputs a reconstructed trajectory. In the next section, we will implement three different types of the map-matching adversaries to carry out simulated attacks.

IV. ALPS: Adaptive Location Privacy-Preserving System with Two-Tier Perturbation

ALPS protects a user's trace privacy by allowing the dynamic obfuscation of a live location trace using a two-tier perturbation process. In the following sections, we provide details of the design and perturbation scheme. In addition, we explain three different adversaries that we have developed for evaluation.

A. Overview

Figure 2 illustrates the architecture of ALPS. Running on a mobile device, ALPS consists of three core components—Synthesizer, Reconstructor, and Evaluator—that has the following salient features:

Use of multiple localization technologies: The effectiveness of perturbation is determined largely by the characteristics of the location sample noise. Measurement error is an *intrinsic noise* in every localization technologies, and can be roughly characterized by an accuracy parameter measured through field survey. ALPS exploits multiple localization technologies readily available on a mobile device, and applies measurement errors from chosen technology for perturbation. In this paper, we consider three localization technologies: GPS, Wi-Fi Positioning System (WPS), and Cellular-ID Lookup (CID), to offer an estimated 10-meter, 100-meter, and 500-meter accuracy, respectively. There are cases where accuracy may vary drastically from general estimated trend, but they are statistically rare and transient thus can be tolerated by our two-tier perturbation scheme and feedback mechanism.

Two-tier perturbation: The Synthesizer in Figure 2 is responsible for generating a perturbed trace. Although perturbing with localization error sounds intuitive and feasible, as pointed out in [9] and our analysis in Section V, perturbation applied independently to individual location sample may be proved vulnerable against adversaries (e.g., outlier filtering). Additional mechanism is required to restore the auto-correlation within the perturbed trace without violating the mobility and topology constraints. Based on this observation, we propose a two-tier perturbation scheme. For each location sample, a first separation tier chooses one localization technique of the multiple ones in a probabilistic manner and outputs the perturbed sample. A second conformation tier then aligns the sample with previous samples in the trace, according to mobility and topology constraints.

Online privacy evaluation and feedback: A mobile user might want to see her level of privacy threat given the perturbed trace against a certain adversary. The Reconstructor and Evaluator in ALPS provide an accurate account on the distortion generated by the Synthesizer over a certain period of time. The Reconstructor generates the reconstructed trace according to an adversary model. It also allows a user to implement and plug-in a new adversary model, as more powerful algorithms become available. The Evaluator then takes the reconstructued trace and calculates its distortion from actual trace (i.e., ground truth). Note that the actual trace can be obtained from underlying localization techniques with the help of the energy-efficient scheme introduced in [10]. Finally, the Evaluator uses this distortion score to evaluate the perturbation performance and provide feedback for parameter setting at Synthesizer.

In summary, when a LBS request is scheduled, the synthesizer makes a decision about perturbation parameter for current location by consulting user-defined privacy profile and history evaluation feedback, and perturbs location sample accordingly to generate a perturbed trace. This perturbed trace is then reconstructed and evaluated to yield a distortion score intuitive enough for user to understand and assess present privacy risk.

B. Two-Tier Perturbation Scheme

Here, we elaborate the detailed operation of the proposed two-tier perturbation scheme in Synthesizer.

1) Separation Tier: The separation tier (Tier-S) takes advantage of the dynamic and varying accuracy guarantees provided by different localization technologies. For each location measurement, one localization technology is chosen for the perturbation. Therefore, the extent of perturbation at a macro scale can be controlled with a probabilistic parameter, denoted as Proportional Parameter (PP). PP is a set of probabilities specifying the fraction that each localization technology is drawn as perturbation source. For instance, a PP of 20/40/40 suggests for current location measurement, a 20% chance of obtaining location fix from GPS as perturbed sample, while 40% chance from WPS and CID equally. Stringent privacy requirement dictates a higher probability of drawing location from less accurate provider such as CID. When requirement is shifted from privacy to QoS, accurate provider like GPS would be favoured.

The probabilistic nature of PP introduces another level of randomness into the perturbation, making it hard for an adversary to generate an accurate estimate of the noise distribution, even with war-driving data from all the localization technologies. Without the exact knowledge about the PP, the best guess that an adversary can make about the perturbation is the zero-mean normal distribution with standard deviation bounded by average accuracy of the least accurate localization technique. Furthermore, the PP can be applied to a per-sample basis, meaning it is flexible enough to support personalized privacy requirement. Within a trace, user can adaptively adjusted PP to maintain the balance between privacy and QoS, according to his privacy profile and accumulative feedback from trace history. Even when there is a conflict between QoS and privacy requirement, some accurate locations may still be published in a privacy-preserving manner by manipulating the perturbation of preceding and succeeding samples.

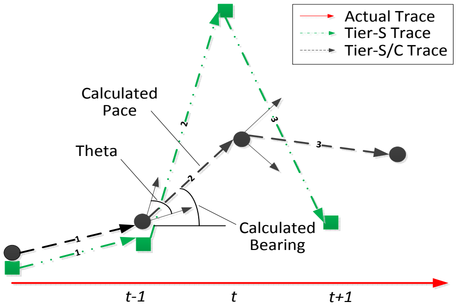


Fig. 3. Pace and Bearing Calculation

2) Conformation Tier: In practice, the gaps of accuracy and granularity across different localization technologies could be large. Applying an independent perturbation scheme to an individual location sample may produce the trace with unrealistic pulse, freezing and bouncing. Adversary equipped with mobility information may then have a better chance of estimating the PP in use. This may help the adversary filtering out such outlier samples to achieve a more accurate reconstruction, as well as making a more informed assessment about the users privacy profile.

Based on this observation, we develop a conformation tier (Tier-C) built upon the separation tier, to smoothen the perturbed trace. This conformation tier reintroduces an artificial correlation that resembles one from a typical mobility scenario between heavily perturbed samples, while preserving the original correlation as much as possible when the perturbation is minor. An example is shown in Figure 3. The solid straight line (in red) denotes the actual trace. The squares (in green) are outputs from the separation tier, while solid circles (in black) demonstrate the operation of conformation tier. First, two samples will be directly taken from Tier-S. To create the conformed sample at time instance $t \geq 2$, denoted as L_{tc}^{t}, the conformation tier uses the perturbed samples from separation tier L_{ts}^{t} as seed, together with previous perturbed samples, L_{ts}^{t-1} and L_{tc}^{t-1} at both tiers. With this information, travel distance (i.e. *Pace*) and turning direction (i.e. *Bearing*) from L_{ts}^{t} are calculated to determine L_{tc}^{t}, as follows:

Pace Calculation: Pace Calculation (PC) algorithm generates the pace from L_{tc}^{t-1} to L_{tc}^{t}, denoted as $P_{tc}^{\Delta t}$. For each interested area, PC will assign a default pace, P_{avg}, calculated using average speed of the area.

First, travel distances from L_{ts}^{t-1} to L_{ts}^{t}, $D_{ts}^{\Delta t}$, is derived. $D_{ts}^{\Delta t}$ is significant in the context of personalized privacy. Since short distances and dense samples in actual trace suggests dwelling and potential sensitivity, separation tier removes this trait in sensitive areas by making larger and more frequent perturbation, while for insensitive areas it remains intact. This property should still be preserved at conformation tier. A threshold $T_{Density}$, defined by user's privacy preference about the general size of sensitive area as well as his average mobility pattern in the past, is used to distinguish the density difference. When $D_{ts}^{\Delta t}$ exceeds the threshold, which suggests that user is

Algorithm 1 Pace Calculation

```
INPUT: Previous Tier-S sample $L_{ts}^{t-1}$; Previous Tier-C sample $L_{tc}^{t-1}$; Current Tier-S sample $L_{ts}^{t}$; Density threshold $T_{Density}$; Maximum distance limit threshold $T_{DL}$; Average travel distance $P_{avg}$
OUTPUT: Expected pace $P_{tc}^{\Delta t}$
$D_{ts}^{\Delta t} = L_{ts}^{t-1}.distanceTo(L_{ts}^{t})$;
$D_{tc\rightarrow ts}^{\Delta t} = L_{tc}^{t-1}.distanceTo(L_{ts}^{t})$;
$P_{tc}^{\Delta t} = D_{tc\rightarrow ts}^{\Delta t}$;
if $D_{ts}^{\Delta t} > T_{Density}$ then
    $P_{tc}^{\Delta t} = P_{avg}$;
else
    if $D_{tc\rightarrow ts}^{\Delta t} < T_{DL}$ then
        $P_{tc}^{\Delta t} = D_{tc\rightarrow ts}^{\Delta t}$;
    else
        $P_{tc}^{\Delta t} = P_{avg}$;
    end if
end if
```

in his normal movement or in sensitive area but perturbed by Tier-S, PC assigns P_{avg} to $P_{tc}^{\Delta t}$.

If $D_{ts}^{\Delta t}$ is below the threshold, which suggests a possible stay in insensitive area, PC calculates travel distances from L_{tc}^{t-1} to L_{ts}^{t}, $D_{tc\rightarrow ts}^{\Delta t}$. A maximum distance limit threshold T_{DL} is defined according to average speed limit in that area. If T_{DL} is not violated, $D_{tc\rightarrow ts}^{\Delta t}$ is assigned to $P_{tc}^{\Delta t}$, in order to converge rapidly to Tier-S samples or to remain close to them. Otherwise P_{avg} is assigned. A formalization of the pace calculation is shown in Algorithm 1.

Bearing Calculation: Bearing Calculation (BC) algorithm generates the bearing from L_{tc}^{t-1} to L_{tc}^{t}, denoted as B_{tc}^{t}. BC records bearing from previous step B_{tc}^{t-1}, and calculates new bearing $B_{tc\rightarrow ts}^{t}$, from L_{tc}^{t-1} towards L_{ts}^{t}. The turning angle (θ) between B_{tc}^{t-1} and $B_{tc\rightarrow ts}^{t}$ will determine the expected bearing B_{tc}^{t}. The principle behind BC is to make turning at each sample as smooth as possible. The algorithm will substitute sharp turns or even u-turns caused by Tier-S perturbation or measurement error, with a sequence of 90°-turns which is more common and realistic in road networks. Based on this, If θ is greater than 90° the algorithm reduces it to 90°. Otherwise, it use half of θ to smoothen the turn. The algorithm is formalized in Algorithm 2.

C. Map-Matching Adversary

As ALPS is flexible and capable of plugging in any adversary model (shown in Figure 2) for the Reconstructor and Evaluator, we have developed three adversaries based on map matching algorithms. The scheme and capability of a map-matching adversary depend on the side-information available. In this section, we implement and study three adversaries assuming different side-information, in order to later demonstrate how significant the context information can improve adversary's ability. All side information used can be extracted from publicly available databases such as OpenStreetMap [11].

1) Nearest-Road (NR) Adversary: The NR adversary is a basic map matching technique that is good at handling samples with small error and relatively simple road topology. The NR adversary does not require much side-information other than a map of roads. Given a perturbed coordinate, the NR adversary simply snaps it onto the nearest road segment. Intuitively, this approach will not perform well at distortion reduction, when the perturbation level is high. We have implemented the NR adversary using the "Get Direction" function provided by Google Maps [12].

Algorithm 2 Bearing Calculation

```
INPUT: Previous Tier-S sample $L_{ts}^{t-1}$; Previous Tier-C sample $L_{tc}^{t-1}$; Current Tier-S sample $L_{ts}^{t}$; Invalid angle constant $B_{null}$
OUTPUT: Expected bearing $B_{tc}^{t}$
INIT: $B_{tc}^{t-1} = B_{null}$
$B_{tc\rightarrow ts}^{t} = calcBearing(L_{tc}^{t-1}, L_{ts}^{t})$;
if $B_{tc}^{t-1} == B_{null}$ then
    $B_{tc}^{t} = B_{tc\rightarrow ts}^{t}$;
else
    $\theta = B_{tc}^{t-1} - B_{tc}^{t}$;
    Normalize $\theta$ within $[-\pi, \pi]$
    if $abs(\theta) \geq \pi/2$ then
        if $\theta < 0$ then
            $B_{tc}^{t} = B_{tc}^{t-1} + \pi/2$
        else
            $B_{tc}^{t} = B_{tc}^{t-1} - \pi/2$
        end if
    else
        $B_{tc}^{t} = B_{tc}^{t-1} - \theta/2$
    end if
    Normalize $B_{tc}^{t}$ within $[-\pi, \pi]$
end if
$B_{tc}^{t-1} = B_{tc}^{t}$;
```

2) Distance-Limit (DL) Adversary: The DL adversary applies a travel distance filter before performing the nearest road map matching. Knowledge of the mobility pattern is used to determine the travel distance threshold T_{DL}^{Adv}. Given the perturbed trace, the DL adversary examines the samples, one by one, based on the threshold; if the travel distance from preceding sample exceeds the T_{DL}^{Adv}, a sample is considered an outlier that is produced by perturbation and, then it is replaced with an interpolated sample.

3) Hidden-Markov-Model (HMM) Adversary: The HMM adversary is a more sophisticated scheme that it considers not only the travel distance limit, but also the road network that regulates the user trajectory. As the perturbation algorithm may introduce relatively large and deviating errors, the HMM adversary adopts similar techniques described in [1] to handle the large error. To our best knowledge, this is the first effort to build an HMM adversary using a map-matching algorithm, and we will detail its design and components below:

- *Area System*: The interested region is gridded into an area system. Each area is a square with the edge length approximated by the average speed and sampling interval, for instance the travel distance during the sampling interval at average speed. Upon the creation of area system, the perturbed trace will first be converted into a sequence of areas.
- *Transition Probability Matrix*: The transition probability matrix models the road topology. Given the area system and road information extracted from map, the HMM first checks the viability of transition between areas, and then stores this information as a transition count (C_{ab}^{T}). Value 0 is assigned if there is no path connecting them, 2 if there is a direct path and 1 if the areas can be connected by a sequence of paths. After creating a transition count matrix, the transition probability for each area can then be calculated as $Prob^{T}(a,b) = \frac{C_{ab}^{T}}{Dist_{M}(a,b)}$, where $Prob^{T}(a,b)$

denotes the transition probability from area a to area b; C^T_{ab} denotes their transition count; $Dist_M(a,b)$ is the Manhattan distance between them. The rationale for this approach is that the transition probability is greater if areas are closer to each other and are connected with fewer number of paths. All calculated transition probabilities are then normalized and recorded in the transition probability matrix.

- *Emission Probability Matrix*: The emission probability describes the likelihood of a location sample from an area being observed at all possible areas. Without knowledge of which particular localization technology is used as well as of corresponding accuracy information, the best an adversary can do is to model the emission probability using normal distribution $\mathscr{N}(0,\sigma_a^2)$ with respect to distance, where σ_a specifies the accuracy estimation. We simplify the calculation by considering Manhattan distance, and choosing an average $\sigma_a = 250$. Therefore we have $Prob^E(md) = \int_{-R(md)}^{R(md)} \frac{e^{\frac{-x^2}{125000}}}{250\sqrt{2\pi}}\,\mathrm{d}x$, where md is the Manhattan distance; $R(\cdot)$ is a function which approximates Manhattan distance to Euclidean distance based on area size.
- *HMM and Viterbi Algorithm*: Given the transition and the emission probabilities, we use Jahmm [13] to generate the HMM model. The observed area sequence and the HMM model are then used as input of the Viterbi algorithm [14] to produce a reconstructed area sequence that has maximum likelihood to be the actual trace.
- *Road Mapper*: The output of Viterbi algorithm is a sequence of areas and needs to be converted into sequence of coordinates. The road mapper finds the representative coordinates for each area in the sequence, by first converting the area sequence to a sequence of road segments that have the smallest number of segments. As each area is matched to a road segment, the road mapper then examine all coordinates traversed by this road segment. The coordinates that are closest to the corresponding coordinate in perturbed trace are used to present the area in the reconstructed trace. Note that this simplified approach may introduce errors depending on the detail of map data. In our experiments, the average of error is about 40 metres.

V. Evaluation

To evaluate the performance of our proposed scheme, several experiments are carried out on realistic traces. For the sake of simplicity, we performed trace-based evaluations. However, the proposed scheme can be easily extended to online cases.

A. Experimental Setup

1) Data Collection: To collect real-life trace data, we developed a location sampler application that runs on the Android platform and obtains location information from GPS, WPS, and CID. The location sampler listens to the Android system‘s location providers and extracts GPS and WPS location updates. It also fetches the user‘s current cellular association information and queries Google Maps API [12] to acquire the corresponding coordinates of associated cellular tower. Note that WPS of Android system provides a hybrid service that automatically switches underlying localization scheme to cellular triangulation when WPS is not available. As a result, an accuracy filter is applied to samples from WPS that record the sample as coming from CID if the accuracy is over 1000 meters. This filter value was set based upon our own empirical data analysis. When the location sampler starts running, it will periodically sample and record the user‘s current location, simulating a typical location tracking and sharing LBS. The sampling interval in our experiment is set to 10 seconds.

Trip	Duration (Minute)	Length (Mile)	Sample Number	Avg. Accuracy (Meter)	
				WPS	CID
Davis	21	3	130	112	516
MV	18	3	110	257	547

TABLE I
Trace Data

In our experiments, the mobile user carries an Android smartphone that is running the location sampler. Location samples are collected when the user moves.

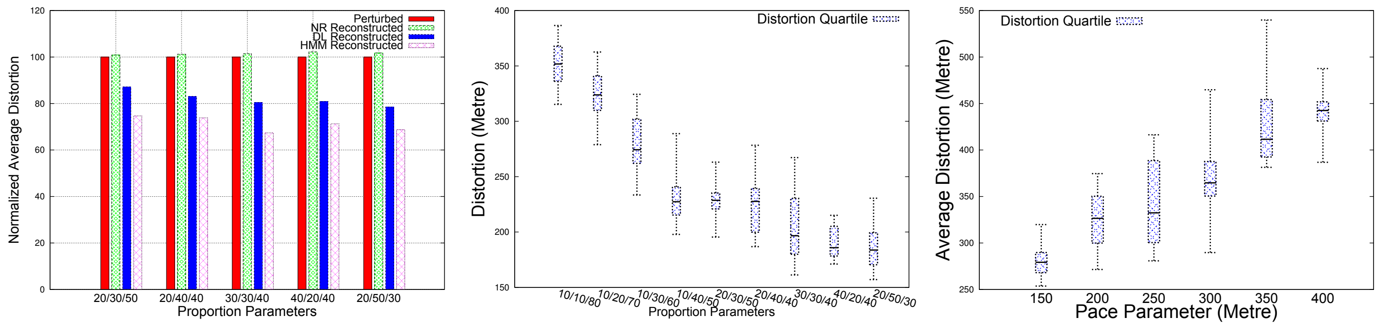
- Davis Trace: The David trace is collected while using the E-Line of Unitrans, a bus service in Davis, CA. The bus drives through campus, commercial, and residential areas, making several stops and traveling at approximately 25 mph for 30 minutes. We defined two sensitive areas with a radius of 100 meters that are centered at the origin and the end of the trace. The rest of the areas in the trace are set as insensitive.
- Mountain View (MV) Trace: The Mountain View trace is collected while driving in a car from the central area of Mountain View, CA. The car drove through commercial and residential areas at an approximately 35 mph for a duration of 18 minutes. Two sensitive areas with a radius of 200 meters are defined.

While the areas in both traces are well covered by WiFi and Cellular networks, null or erroneous samples were caused by a lack of localization coverage. These samples were removed to improve visualization. GPS trace is used as ground truth with no error. We summarized the characteristics of the traces in Table I.

2) Perturbation Schemes and Adversaries: In addition to the two-tier perturbation scheme, we implemented three additional perturbation schemes for the purpose of comparison as follows:

- Gaussian perturbation: Gaussian scheme used in [15] perturbs location sample independently by introducing a displacement that is created with (1) a turning direction, uniformly chosen from $[-\pi,\pi]$, and (2) a perturbing distance that follows I.I.D Gaussian distribution $N(0,\sigma_p^2)$. Level of perturbation is controlled with parameter σ_p.
- Tier-S perturbation: Tier-S scheme perturbs samples with the separation tier mechanism only. The perturbation can be controlled by the proportional parameter (PP).
- Tier-C perturbation: Tier-C scheme perturbs samples with conformation tier mechanism only, in which case separation tier always supply non-perturbed samples from GPS. The perturbation can be controlled by the pace parameter (PC).

In addition, we implemented three adversaries (NR, DL, HMM), described in Section IV. Table II summarizes their default parameters.

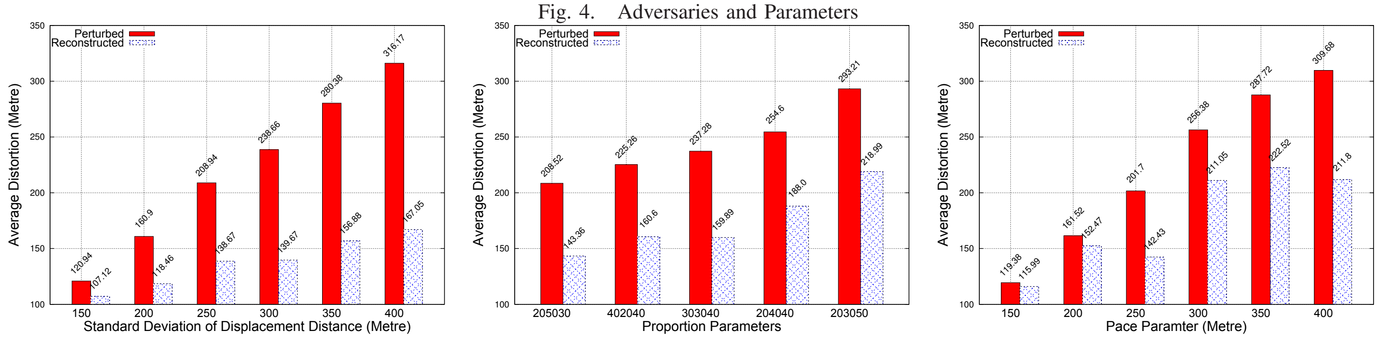


(a) Adversary Capability

(b) Effect of PP

(c) Effect of PD

Fig. 4. Adversaries and Parameters

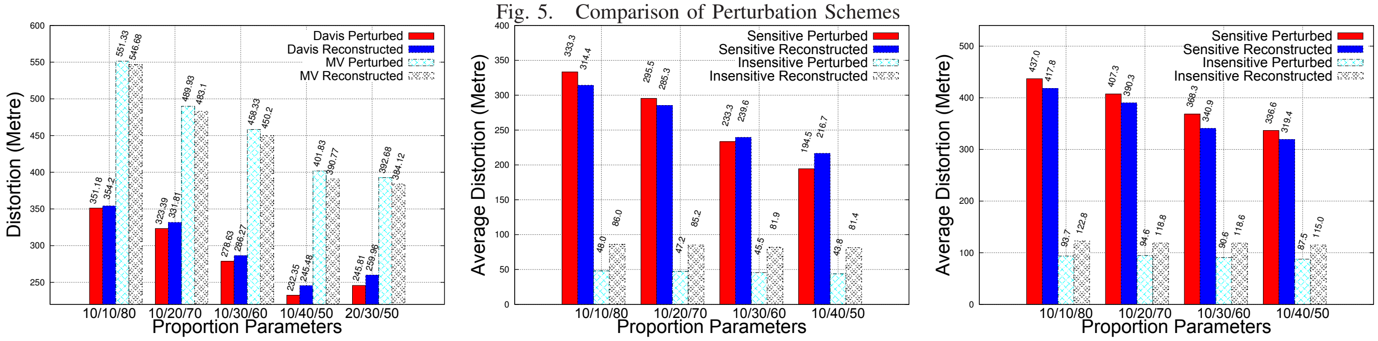


(a) Gaussian Perturbation

(b) Perturbation Tier

(c) Conformation Tier

Fig. 5. Comparison of Perturbation Schemes



(a) Average Distortion on Two Traces

(b) Personalized Privacy: Davis

(c) Personalized Privacy: Mountain View

Fig. 6. Overall Performance of Two-Tier Perturbation

3) Metric: The distortion metric described in Section III-B is used to quantify both privacy and QoS. Large distortion values indicate a high level of privacy, but a reduced QoS level. The distortion difference between a perturbed trace and a reconstructed trace shows the effectiveness of the perturbation scheme for its defense against the adversary.

B. Experiment Results

We first conduct several microscopic experiments to examine the feasibility of the scheme and to validate our design choices. Next, we give a macroscopic evaluation on the performance of the two-tier perturbation scheme, in terms of overall distortion in the presence of an adversary and Privacy/QoS trade-off for personalized privacy.

1) Comparison of Adversary Performance: We first examine the performance of the different map-matching adversaries (NR, DL, HMM) at attacking an uncorrelated perturbation, in order to demonstrate the impact of side information on their reconstruction process. In this experiment, The Davis trace is perturbed by the Tier-S scheme to create an uncorrelated perturbed trace, which is then reconstructed by all the three adversaries. We tested 5 sets of different proportional parameters (PP), and repeated reconstruction 20 times. For each PP, the distortion values are averaged and normalized using the distortion from the perturbed trace as unit metric. Difference between perturbed and reconstructed bars shows us the distortion reduction by adversary.

Trip	$T_{Density}$ (Meter)	T_{DL} (Meter)	P_{avg} (Meter)	T_{DL}^{Adv} (Meter)	Area Size (Meter)
Davis	50	150	110	150	150
Mountain View	50	200	160	200	200

TABLE II
DEFAULT PARAMETER VALUES

Figure 4(a) compares the distortion reduction performance of the three adversaries. As shown in the figure, we can observe that the NR adversary has no effect on the uncorrelated perturbation, since it performs reconstruction based only on minimal geographic knowledge, disregarding the correlation inherited from the user mobility and road topology. On the other hand, the DL adversary, which takes mobility constraints into consideration, achieves an average 18% (45-meter) reduction in the distortion. Finally, the HMM adversary, with a very coarse estimation of road network topology as well as mobility constraint, achieves a reduction of over 29% (70 meters). As expected, the result confirms that side information can be exploited by adversary to improve her performance in reconstruction.

2) Comparison of Perturbation Schemes: Before we study the performance of the proposed two-tier scheme, we would like to examine several different perturbation schemes, in order to assess the inadequacy of such schemes against sophisticated map-matching adversary and to set up a baseline for justifying and evaluating the two-tier scheme. In this experiment, we use the Davis trace. We apply the Gaussian, Tier-S, and Tier-C perturbation schemes to the trace and test thus-perturbed trace against a HMM adversary. We vary σ_p value to change the level of Gaussian perturbation.

Figure 5(a), Figure 5(b), and Figure 5(c) show the average distortion reduction of HMM adversary on the three schemes respectively. As shown in the figures, significant distortion reduction can be observed from Figure 5(a). This reaffirms that I.I.D Gaussian perturbation is not an effective perturbation scheme against powerful adversary. Next, in the test of Tier-S scheme, multiple sets of PP are used to provide different level of perturbation. From the figure, we still observe large distortion reductions. Tier-S perturbation partially destroys the correlation between samples, by probabilistically choosing inaccurate localization technologies. Therefore, if an adversary can find a way to estimate and restore the correlation, the distortion can still be reduced.

In contrast to Gaussian and Tier-S scheme, Tier-C scheme is used to re-introduce the correlation into perturbed trace that complies with mobility and topology constraints. The key is to create an artificial correlation with trait similar to the one in the actual trace. Distortion and similarity in correlation are both largely controlled by pace parameter. We increase the pace from 150 meter to 400 meter, at a step of 50 meter. From Figure 5(c), it is intuitive to see that the distortion increases as longer pace parameter is assigned. However, we can also observe that when pace parameter is over-scaled to an extent, which is certainly unrealistic and impossible to reach by mobile user, adversary successfully captures and exploits this property for distortion reduction. The result suggests that it is difficult for Tier-C scheme to offer adequate distortion and simulate plausible correlation at the same time.

3) Effect of Parameters on Privacy Control: The two-tier scheme controls perturbation mainly through two parameters, the proportional parameter at separation tier and the pace parameter at conformation tier. In this experiment we design two sets of tests to examine the effect of these two parameters in detail. The Davis trace is used.

In the first set of tests, we provide the separation tier the different combination of proportional parameters, with fixed pace parameter of 110 meters. Figure 4(b) shows the quartile statistics of distortion created by the two-tier scheme. From the figure, one can observe clearly that distortion increases as the proportion of CID samples increases. It is less obvious but still a general trend that increasing the proportion of GPS samples may reduce the standard deviation in distortion. Nonetheless, the result suggests that the scheme can effectively adjust the level of perturbation by rearranging the proportional parameter.

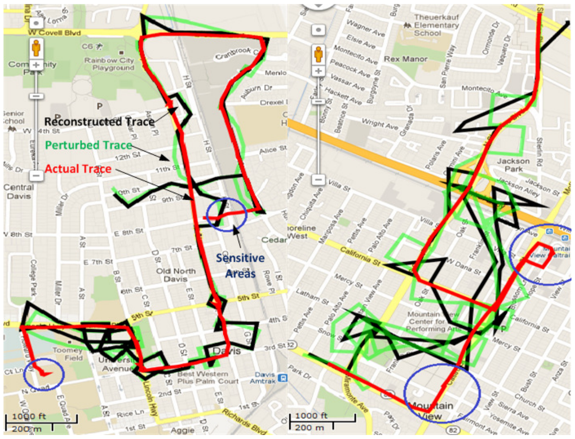


Fig. 7. Personalized Privacy on Two Traces

As shown in the last experiment, the pace parameter is generally associated with user mobility and, therefore, it might be less favorable parameter for controlling the privacy. However, in some situations, it is possible for the scheme to dynamically change the pace parameter, in order to further confuse an adversary. In this test, we fix the PP at 10/30/60 and change pace distance from 150 to 400 meters, at a step of 50-meter. Note that the choice of PP won't affect the trend of results, and we use 10/30/60 as default to represent a balance between privacy and energy saving. The result is shown in Figure 4(c). As expected, the pace parameter exhibits a positive correlation with the distortion. Also, note that distortion is multiplied by the two-tier scheme, comparing to the previous Tier-C scheme results shown in Figure 5(c).

4) Performance of Two-Tier Perturbation: We then examine the overall distortion of the proposed two-tier perturbation scheme against various adversaries. In addition, we evaluate its performance when personalized privacy level is specified. Evaluations are carried out for both Davis trace and Mountain View traces. Here, only the proportional parameter is used to adjust distortion level, and the pace parameter is set to its default value. Only results from HMM adversary are shown due to limitation of space.

Figures 6(a) shows the average distortion on both traces produced by the two-tier scheme, before and after the reconstruction. The first thing we observed is that there is no apparent distortion reduction, suggesting that the effect of attack is negligible, and the scheme is resilient to these adversaries. Also, note that with the help of the conformation tier, the scheme can afford to use the proportional parameter that heavily favors CID when privacy requirement is high. At the same time, it helps in avoiding freezing or bouncing in the perturbation, making the trace more realistic.

The two-tier scheme applies personalized privacy by assigning different proportional parameters (PP) according to the location sensitivity. We now study this transition between different PP settings within one trace. For samples that fall into sensitive regions, PP with large CID fraction is assigned to ensure privacy required. Otherwise, PP with 100% GPS is used to maximize QoS.

Figures 6(b) and 6(c) show the average distortion for sensitive and insensitive samples and their performance against HMM adversary. As shown in the figures, we observe that desirable distortions are achieved. For samples within sensitive regions, high level of distortion is maintained, while for samples outside the areas, only a relatively small amount of distortion is introduced. More importantly, we can see that even the powerful adversary cannot effectively reduce the distortion through their reconstruction. Figures 7 gives an illustration of this effect on both traces: Davis on the left and Mountain View on the right. The red, green, and black trajectory represents actual, perturbed and reconstructed trace respectively. Blue circle suggests sensitive regions. We see adversary is led astray from sensitive regions in a natural manner. When user move out the sensitive regions, perturbed trace is quickly converged. This result suggests personalized privacy requirement for trace can be achieved with comparable privacy and QoS guarantees.

5) Discussion: The synthesizer doesn't involve any computational intensive operations which allows it to work as a middleware between OS location providers and other APPs. By accommodating multiple location providers for perturbation, frequent need of sampling with power intensive localization technology can also be reduced. In the end, single perturbed sample is sent to LBS provider thus no additional communication overhead incurred due to LPPM.

VI. Related Work

There are many related studies in the location privacy. In [2], the authors summarize LBS related privacy issue into two major categories based on the objective: communication privacy (identity) and location privacy (whereabouts). Numerous location privacy preservation mechanisms (LPPMs) [16], [17] have been proposed. In [18], LPPMs are formalized into four primary methods: Hiding Events, Adding Dummy, Obfuscation and Anonymization. Obfuscation [19], including perturbation [20], [15] and cloaking [21], is a popular method that can be used to protect both location privacy and anonymity.

Privacy metrics are also important to evaluate the performance of LPPMs, and several metrics such as *k*-anonymity and entropy-based [22] have been proposed. Work in [23], however, pointed out that such metrics may not be able to truthfully report the status of location privacy, and thus the concept of *l*-diversity is introduced. By considering the adversary knowledge and potential attacks [24], [8], distortion-based metric is also proposed.

As anonymity of location trace has been heavily studied, location privacy of continuous location updates also attracts more and more attention [25], [3]. Meanwhile, advanced techniques developed for map matching task [26], [27], [1] open the chance for the development of more powerful adversaries.

VII. Conclusion

This paper has presented ALPS that provides an adaptive location-tracking privacy preservation scheme with a mobile user. ALPS allows a user to understand the threat of location trace privacy and to control their privacy level, depending on the sensitivity of location. ALPS esentially takes the two-tier approach, through which the system can introduce different perturbation degrees to the ordinal trace and, at the same time, can conform the thus-perturbed traces by adding meaningful correlation to be resistant against powerful adversary. To evaluate ALPS, we have identified and developed three adversaries that are based on map-matching algorithms. We have implemented and evaluated ALPS extensively by using real-life traces that we collected over campus and urban areas, and demonstrated the ALPS' effectiveness and controllability of location privacy preservation against the adversaries.

References

[1] A. Thiagarajan, L. S. Ravindranath, H. Balakrishnan, S. Madden, and L. Girod, "Accurate, Low-Energy Trajectory Mapping for Mobile Devices," in *8th USENIX NSDI*, Boston, MA, March 2011.

[2] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proc. of MobiSys*, New York, NY, USA, 2003, pp. 31–42.

[3] G. Ghinita, M. L. Damiani, C. Silvestri, and E. Bertino, "Preventing velocity-based linkage attacks in location-aware applications," in *Proc. of GIS*, 2009, pp. 246–255.

[4] J. Krumm, "Realistic driving trips for location privacy," in *Proceedings of Pervasive Computing*, 2009, pp. 25–41.

[5] H. Kido, Y. Yanagisawa, and T. Satoh, "An anonymous communication technique using dummies for location-based services," in *Proc. of ICPS '05*, july 2005, pp. 88 – 97.

[6] "Google Latitude," http://www.google.com/latitude/.

[7] M. Duckham and L. Kulik, "Location privacy and location-aware computing," *Dynamic and Mobile GIS: Investigating Change in Space and Time*, 2006.

[8] R. Shokri, J. Freudiger, M. Jadliwala, and J.-P. Hubaux, "A Distortion-based Metric for Location Privacy," in *Proc. of WPES*, November 2009.

[9] N. Pham, R. K. Ganti, Y. S. Uddin, S. Nath, and T. Abdelzaher, "Privacy-preserving reconstruction of multidimensional data maps in vehicular participatory sensing," in *Proc. of EWSN*, 2010, pp. 114–130.

[10] J. Paek, K.-H. Kim, J. Singh, and R. Govindan, "Energy-efficient positioning for smartphones using cell-id sequence matching," in *ACM MobiSys '11:*, 2011.

[11] "Open Street Map," http://www.openstreetmap.org/.

[12] "Google Maps API," http://www.google.com/glm/mmap.

[13] "Jahmm," https://code.google.com/p/jahmm/.

[14] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257 –286, feb 1989.

[15] J. Krumm, "Inference attacks on location tracks," in *Proceedings of*, ser. PERVASIVE'07, 2007, pp. 127–143.

[16] K. G. Shin, X. Ju, Z. Chen, and X. Hu, "Privacy Protection for Users of Location-Based Services," *IEEE Wireless Communication*, vol. 19, no. 1, pp. 30–39, 2012.

[17] J. Krumm, "A survey of computational location privacy," *Personal Ubiquitous Comput.*, vol. 13, no. 6, pp. 391–399, Aug. 2009.

[18] R. Shokri, J. Freudiger, and J. P. Hubaux, "A unified framework for location privacy," in *Proc. of PETS*. Citeseer, 2010, pp. 203–214.

[19] M. Duckham and L. Kulik, "A formal model of obfuscation and negotiation for location privacy," in *Proceedings of*, ser. PERVASIVE'05, 2005, pp. 152–170.

[20] B. Hoh and M. Gruteser, "Protecting location privacy through path confusion," in *Proceedings of*, ser. SECURECOMM '05, 2005, pp. 194–205.

[21] M. Damiani, C. Silvestri, and E. Bertino, "Fine-grained cloaking of sensitive positions in location-sharing applications," *Pervasive Computing, IEEE*, vol. 10, no. 4, pp. 64 –72, april 2011.

[22] A. Beresford and F. Stajano, "Location privacy in pervasive computing," *Pervasive Computing, IEEE*, vol. 2, no. 1, pp. 46 – 55, jan-mar 2003.

[23] M. Xue, P. Kalnis, and H. K. Pung, "Location diversity: Enhanced privacy protection in location based services," in *Proceedings of*, ser. LoCA '09, 2009, pp. 70–87.

[24] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux, "Quantifying location privacy," in *Proceedings of the 2011 IEEE Symposium on Security and Privacy*, 2011, pp. 247–262.

[25] M. Gruteser and X. Liu, "Protecting privacy, in continuous location-tracking applications," *Security Privacy, IEEE*, vol. 2, no. 2, pp. 28 – 34, mar-apr 2004.

[26] J. Krumm, J. Letchner, and E. Horvitz, "Map Matching with Travel Time Constraints," in *Society of Automotive Engineers (SAE) 2007 World Congress*, Apr. 2007.

[27] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson, "Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones," in *ACM SenSys '09*, 2009, pp. 85–98.