A Novel Algorithmic Structure for Concealing the Location and Path Movement of a Mobile Device

Kianoosh G. Boroojeni, Sebastian Zanlongo, S. S. Iyengar, *Fellow, IEEE,* and Niki Pissinou, *Fellow, IEEE*

***Abstract*—One of the increasingly important challenges in location-based services is how to preserve the location privacy of users while providing different services based on their location. In this paper, we propose a novel stochastic mechanism to preserve the location privacy of a mobile node randomly walking in an Euclidean plane by obfuscating its instantaneous location and concealing its path movement (track). The mechanism enables the node to specify the minimum level of privacy ($\lambda \in (0, 1)$) that it desires or the maximum error tolerance (ε) that it is willing to accept when informing the services of its location. We quantify the privacy level using the expected distortion of the adversary's guess from the device location. Additionally, the trade-off between the error tolerance and privacy level will be specified. Our experimental result supports the theoretical analysis of our framework. To the best of our knowledge, this is the first paper which finds a theoretically proven lower-bound on the location privacy of a user with random walk.**

***Index Terms*—location privacy, mobile sensor, random walk**

I. Introduction

RECENT advances in positioning technologies (*e.g.* GPS) and mobile devices like wireless sensors and smart phones have increased the demand for Location-Based Services (LBS's). One major concern in the wide deployment of LBSs is how to preserve the location privacy of users while providing them with a service based on their locations. LBS providers can be victims of the privacy attackers (adversary) to track the users, or they themselves may abuse the users' location information for different purposes (for example, some mobile apps abuse their users' location for advertising purposes) [1], [2].

A. Motivation

Consider a mobile sensor in a Wireless Sensor Network (WSN) which moves over an area to examine some criteria. To do this, it goes to different places and frequently uses some LBS to efficiently choose its path and get specific information regarding its current location. Our focus in this paper will be on the employing of LBS's in the context of sensor networks like Mobile Ad hoc NETworks (MANETs). Let's assume that the sensor *randomly walks* over an area to examine some criteria in different places. There are various reasons that makes the sensor randomly walk. One of them is that moving regularly degrades the quality of sensing. For example, assume that a mobile sensor is designed to move over an area and examine the transmitted packets in a Wireless Sensor Network (WSN) to check if there exists some data security attack in different points of the network. If the sensor walks regularly (not randomly), the attacker will learn the mobility pattern of the controlling sensor. Then, based on this information, he only attacks when it is unlikely that the controlling sensor approaches them.

So, in some WSN applications, random walk is applied in order to protect location privacy. However, as the random motion sensors find a need to expose their location to some LBS, their location privacy may then be threatened by the LBS itself or some third party (adversary). In addition to this primary issue, recently, Shi *et al.* [3] showed that random walk doesn't provide a true realistic memory-less structure. Based on their work, merely using a random walk doesn't guarantee the preservation of location privacy as the movement path will help the adversary to predict the future motions. These couple of issues make a compelling reason to design a smart security mechanism for randomly walking devices to hide their *instantaneous location* and also preserve the current *path movement* structure to conceal the unexpected future paths of a mobile device.

B. Related Work

There are two general approaches dealing with the location privacy issue in LBSs. In this section, firstly we address the couple of approaches; then, we explain how to quantify the location privacy of LBS users. Finally, we will focus on our contribution.

k-Anonymity Cloaking: In this approach, instead of sending one single user's LBS request to the server, including her exact location, k-anonymity cloaking employs a trusted third-party who collects k neighboring users' requests and sends them all together to the LBS service provider. This approach doesn't address the case that the user density is high; in this csae, the k users' locations may be very close to each other, and hence this approach will still reveal the user's location privacy to some extent. This approach was originally proposed by Gruteser and Grunwald [4]. Their work may lead to large service delay if there are not enough users requesting LBSs. Later on, Gedik and Liu [5] designed a joint spatial and temporal cloaking algorithm which collects k LBS requests, each from a different user in a specified cloaking area within a specified time period and then sends them to the service provider. A negative point in their work is that if there are only less than k requests within the predefined time period, the users' requests will be blocked.

In 2009, Meyerowitz and Choudhury [6] tried to improve the service accuracy by predicting the users' paths and LBS

The authors are with the School of Computing and Information Sciences, Florida International University, Miami, FL, 33199, USA.
e-mail: {kghol002,szanl001,iyengar,pissinou}@fiu.edu

Work	Year	Mobility Pattern	LPPM[a]	Privacy Quantification[b]	Frequency of Location Exposure[c]
Hoh *et al.* [12]	2007	for real vehicles	Location Obfuscation	No	Continuous
Gong *et al.* [18]	2010	N/A	Anonymization	No	Sporadic
Shokri *et al.* [19]	2011	for real vehicles	Location Obfuscation	Yes	Sporadic
Shokri *et al.* [19]	2011	for real vehicles	Location Obfuscation	Yes	Sporadic
Shokri *et al.* [20]	2012	for real people[d]	Anonymization	Yes	Sporadic
Li *et al.* [2]	2013	N/A	Location Obfuscation	No	Sporadic
This Work	2014	random walk[e]	Location Obfuscation	Yes	Continuous

[a] Location Privacy-Preserving Mechanism

[b] Quantifying the privacy-level as the expected distortion of the adversary's guess.

[c] Frequency of location exposure differs when using different LBSs.

[d] who use various means of transportation

[e] which is applicable in mobile sensors and ad-hoc networks.

TABLE I: Comparison of different works in the context of location privacy-preserving mechanism.

queries, and send the results to users' before they submit queries. The main drawback of their approach is the network delay occurred because of high communication overhead. For more treatment on k-anonymity cloaking, see [7], [8], [9], [10], [11], [12].

Location Obfuscation: We can divide the solutions with this approach into two categories: solutions that preserve the location privacy of the users by inserting some fake LBS requests; and those which deviate a user's location from the real one in her LBS request to protect her location privacy.

As examples of the solutions in the earlier category, consider the schemes proposed by Kido et al. [12], Lu et al. [13], and Duckham and Kulik [14]. In these schemes, the user generates some fake locations (dummies) using some dummy generation methods and submits the dummies and its own location to the LBS server. The server analyzes every submitted query and replies properly. The major drawback of the solutions in this category is that the server is used inefficiently and may become the system bottle-neck. Additionally, users' location privacy is not preserved in advance.

The second category of solutions like the ones proposed by Ardagna et al. [15], Pingley et al. [16] and Damiani et al. [17] hide users' real locations, e.g., by submitting shifted locations. Such schemes trade service accuracy for location privacy.

Finally, in 2013, Ming Li et. al. [2] proposed a location privacy preserving scheme called n-CD which doesn't use a third party and provides a trade-off between the privacy level and the system accuracy (concealing cost).

Location Privacy Quantification and Formalization: In 2011, Shokri et al. [20,21] formalized localization attacks using Bayesian inference for Hidden Markov Processes. They mathematically modeled a location privacy-preserving mechanism, users mobility pattern and adversary knowledge-base. They also divide the LBSs into two classes: those who *sporadically* ask their users to expose their location, and those who *continuously* do. Additionally, they quantified the user location privacy as the expected distortion of adversary's guess from the reality of user's location. They used this quantification method to evaluate the effectiveness of location obfuscation and fake location injection mechanisms in the improvement of location privacy level.

C. Our Contribution

In this paper, we obfuscate the movement path (track) and instantaneous location of a randomly walking mobile device in 2-D plane using a novel mechanism. Our approach enables the user to specify the minimum level of privacy (λ) that it desires and the maximum error tolerance (ε) that it is willing to accept when informing the location based services of its location. The level of privacy is defined as the expected error (distortion) of adversary's guess from the user's location. Moreover, the error tolerance is defined as the expected Euclidean distance between the LBSs' guess and user's actual location. We will find a trade-off between λ and ε which will be examined using some simulation results. Table I shows our contribution in comparison with a number of major works that have been done in the area of designing location privacy-preserving mechanisms.

II. Problem Specification

Let c denote a mobile node in an Euclidean plane ($\mathbb{R}^2$) that wants to use a location-based service (say A). Assume that c is walking on the plane to do some job; for example, let c be a *sensor* which wants to control some criteria on the 2-D plane. Additionally, assume that there is no obstacle on the plane and node c can go anywhere on the plane (we leave the case with obstacles for future work). We discretize the plane by partitioning it into an infinite countable number of congruent distinct regions. As the result, node c is located in one of the regions at any given moment. Let r_t denote a random process which specifies the region in which c is located at moment $t \geq 0$.

Location-Based Service

Assume that node c sends a query message containing its location to location-based service A every one time unit (which can be any value) and gets benefit of its service. Here is the format of the query message that node c sends to A at time $t = n$ (for $n = 0, 1, \ldots$):

$$\mathcal{Q}_c(n) = \langle \mathrm{ID}_c, r_n, \mathrm{DATA} \rangle \tag{1}$$

where ID_c specifies the ID of node c, r_n denotes the region where c is located in at time n, and DATA represents the other information that c may need to send to the LBS.

This kind of communication makes the LBS able to keep track of node c during the communication time. This may compromise its privacy (as the LBS or some third party (like data sniffer) may abuse this information). Consequently, we need to define a Location Privacy-Preserving Mechanism (LPPM) to filter the query message before sending it to the LBS. In this paper, we use a location-obfuscation method to filter this information and increase the privacy level of node c.

Location Privacy-Preserving Mechanism

In order to preserve the location privacy, node c obfuscates its current location using an LPPM before sending it to the LBS. In fact, instead of sending query $\mathcal{Q}_c(n)$ at time $t = n$, it sends $\mathcal{Q}'_c(n)$ such that:

$$\mathcal{Q}'_c(n) = \langle \mathrm{ID}_c, \mathrm{obf}(r_n), \mathrm{DATA} \rangle \tag{2}$$

where obf: $\mathcal{R} \mapsto 2^{\mathcal{R}}$ and $\mathcal{R} = \{r_0, r_1, r_2, \ldots\}$ (note that function obf maps every region to a set of regions). In this way, node c sends an obfuscated location each time instead of revealing its exact location to the LBS. This obfuscated location is obtained by the following equation:

$$\mathrm{obfuscated}(r_t) = \bigcup_{\rho \in \mathrm{obf}(r_t)} \rho \tag{3}$$

We call the sequence of regions $\bar{r} = r_0, r_1, r_2, \ldots$ as the *(actual) track* of node c; while the sequence:

$$\mathrm{obfuscated}(r_0), \mathrm{obfuscated}(r_1), \mathrm{obfuscated}(r_2), \ldots$$

is the corresponding *obfuscated* track of c.

Adversary

As mentioned before, the LBS itself is considered to be a potential adversary. Additionally, some third party may eavesdrop the query messages originated by node c to use them for some malicious purposes. In this paper, we assume that the adversary knows the LPPM (which is function obf) used by node c to filter the query message. Additionally, at time $t = n$, she is aware of the obfuscated track sent by c in time interval $[0, n]$. Using this information, the adversary wants to reasonably guess the actual track of node c. Let $\hat{\bar{r}} = \widehat{r}_0, \widehat{r}_1, \ldots$ denote the adversary guess of the actual track such that for every i, $\widehat{r}_i \in \mathcal{R}$ specifies the adversary guess of region r_i in which node c is located at time $t = i$.

III. The Proposed Mechanism

Here in this section, we describe a novel location privacy-preserving mechanism which obfuscates the location information of mobile node c to increase its privacy. In order to specify such a mechanism, we need to define the output value of the aforementioned function obf for every given input region $r \in \mathcal{R}$.

In this section, we assume that set $\mathcal{R}$ partitions the Euclidean plane into an infinite number of unit squares such that:

$$\forall \rho \in \mathcal{R}, \exists x, y \in \mathbb{R} : \rho = [x, x+1) \times [y, y+1) \tag{4}$$

Additionally, we discretize the time space into the set of non-negative integer (as we have already assumed that the query messages are sent every single time unit).

Algorithm 1 specifies how the mechanism works.

Algorithm 1: RegionObfuscator

```
Input: region $\rho$, time $n$, & scale factor $\alpha$
Output: obfuscated region $o$ & boundary factor $\mathcal{B}$
if $n = 0$ then
    $x \leftarrow \mathcal{U}\mathrm{nif}(-\frac{1}{6}, \frac{1}{6})$; $y \leftarrow \mathcal{U}\mathrm{nif}(-\frac{1}{6}, \frac{1}{6})$;
    $o \leftarrow$ A square of edge length $\alpha$ and
    centroid $\rho$.centroid;
    /* edges are parallel to $x$-$y$ axes */
    $o \leftarrow$ Translation$\big(o, (x\alpha, y\alpha)\big)$;
    $\mathcal{B} \leftarrow \mathcal{U}\mathrm{nif}(\frac{1}{3}, 1)$;
end
else
    $o' \leftarrow$ the obfuscated region of time $n-1$;
    $\mathcal{B}' \leftarrow$ the boundary factor of time $n-1$;
    $\mathcal{S} \leftarrow$ the square of edge length $\mathcal{B}'\alpha$
    and centroid $o'.centroid$;
    if $o \subset \mathcal{S}$ then
        $o \leftarrow o'$;
        $\mathcal{B} \leftarrow \mathcal{B}'$;
    end
    else
        $o \leftarrow$Update$(\rho, o')$;
        $\mathcal{B} \leftarrow \mathcal{U}\mathrm{nif}(\frac{1}{3}, 1)$
    end
end
return $(o, \mathcal{B})$;
```

As you see, function RegionObfuscator gets region $\rho \in \mathcal{R}$, time $t = n$, and scale factor α as its input and calculates the corresponding obfuscated region and *boundary factor* (which will be addressed later) as the output.

In the case that $n = 0$, the obfuscated region is a square of edge length α with parallel edges to x-y axes. The square is centered at a point obtained by randomly translating the ρ's centroid:

$$o.\mathrm{centroid} = \rho.\mathrm{centroid} + (x\alpha, y\alpha) \tag{5}$$

where x and y are uniformly distributed over interval $(-1/6, 1/6)$. Additionally, the boundary factor is obtained by generating a sample of random variable $\mathcal{U}\mathrm{nif}(1/3, 1)$.

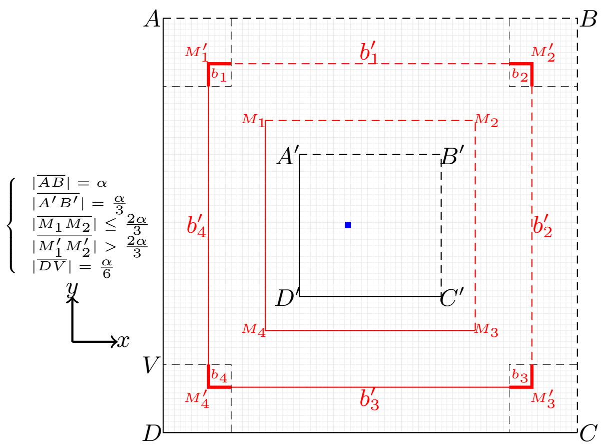
A
B
M_1'
b_1'
M_2'
b_1
b_2
M_1
M_2
A'
B'
$|\overline{AB}| = \alpha$
$|\overline{A'B'}| = \frac{\alpha}{3}$
$|\overline{M_1M_2}| \leq \frac{2\alpha}{3}$
$|\overline{M_1'M_2'}| > \frac{2\alpha}{3}$
$|\overline{DV}| = \frac{\alpha}{6}$
b_4'
b_2'
D'
C'
y
x
M_4
M_3
V
b_4
b_3
M_4'
b_3'
M_3'
D
C

Fig. 1: The schematic illustration of the proposed mechanism. As you see, the 2D plane is partitioned into congruent square-shape regions. The blue square shows the original region (ρ) in which node c is located at time $t = 0$. Square $M_1M_2M_3M_4$ ($M_1'M_2'M_3'M_4'$) specifies area $\mathcal{S}$ for boundary factor $\mathcal{B} = 0.5$ ($\mathcal{B} = 0.78$).

Figure 1 illustrates the obfuscated region (which is $o = \square ABCD$) corresponding to the specified original region (ρ) at time $t = 0$. It is easy to show that region ρ always lies into square $\square A'B'C'D'$ of edge length $\alpha/3$ and centroid $o.centroid$ (see Equation 5).

As you see in Algorithm 1, in the case that $n > 0$, function REGIONOBFUSCATOR first needs to check whether the current obfuscated region and boundary factor (which were generated at time $t = n - 1$) are *valid* for the current region ρ. If yes, it simply returns the previous values; otherwise, it generates a new boundary factor ($\mathcal{B}$) and calls function UPDATE to generate a new region (o).

Here is the validation rule: if region ρ lies inside the square $\mathcal{S}$ (which is centered at o'.centroid and has the edge length of $\mathcal{B}'\alpha$), the current obfuscated region (o') and boundary factor ($\mathcal{B}'$) are still valid at time $t = n$.

Now, we consider the case that the validation rule doesn't hold. In other words, node c is no longer inside square $\mathcal{S}$. Let l denote the line segment connecting points r_{n-1}.centroid and r_n.centroid (note that $\rho = r_n$). Assuming that $\mathcal{B}' > 2/3$, there are two possible cases (regarding Figure 1, consider $\mathcal{S} = \square M_1'M_2'M_3'M_4'$):

Case 1If l intersects b_i, region o will be the area inside a square of edge length α and centroid M_i' (for every $i = 1, 2, 3, 4$).

Case 2If l intersects b_i', region o will be the area inside a square of edge length α and centroid N_i (for every $i = 1, 2, 3, 4$). Assuming that random variable Z is uniformly distributed over interval $(-1/6, 1/6)$, point $N_i = (X_i, Y_i)$ is obtained by the following equations:

$$X_i = \begin{cases} \rho.\text{centroid}.x + \alpha Z & i = 1 \\ o'.\text{centroid}.x + \frac{\alpha}{3} & i = 2 \\ \rho.\text{centroid}.x + \alpha Z & i = 3 \\ o'.\text{centroid}.x - \frac{\alpha}{3} & i = 4 \end{cases} \tag{6}$$

$$Y_i = \begin{cases} o'.\text{centroid}.y + \frac{\alpha}{3} & i = 1 \\ \rho.\text{centroid}.y + \alpha Z & i = 2 \\ o'.\text{centroid}.y - \frac{\alpha}{3} & i = 3 \\ \rho.\text{centroid}.y + \alpha Z & i = 4 \end{cases} \tag{7}$$

In addition, if $\mathcal{B}' \leq 2/3$, (i.e., regarding Figure 1, area $\mathcal{S}$ will be similar to the region inside $\square M_1M_2M_3M_4$), there exists only one possible case (Case 2), as $b_i = \emptyset$ for every $i = 1, 2, 3, 4$.

IV. Analysis of the Privacy Level

In this section, we analyze the location privacy level of our mechanism using the quantification method proposed by Shokri et al. The privacy level is defined as the *expected value of adversary error in guessing the track of node* c. Before presenting the formal definition of privacy level, let's introduce a couple of notations:

$$\begin{cases} \bar{r}_\tau = r_0, r_1, \ldots r_\tau \\ \bar{\hat{r}}_\tau = \hat{r}_0, \hat{r}_1, \ldots \hat{r}_\tau \end{cases} \qquad \forall \tau \in \mathbb{Z}^+$$

where r_i and $\hat{r}_i$ are defined as what mentioned before. The instantaneous privacy level at time τ is obtained by the following equation:

$$\lambda(\tau, \bar{\rho}) = \sum_{\bar{\hat{r}}_\tau} \mathbf{Pr}[\bar{r}_\tau = \bar{\hat{r}}_\tau | \mathcal{K}] \Delta(\bar{\hat{r}}_\tau, \bar{\rho}_\tau) \tag{8}$$

where $\bar{\rho}_\tau = \rho_0, \rho_1, \ldots, \rho_\tau$ denotes the values of process r_t at times $0, 1, \ldots, \tau$. Additionally, function $\Delta(\bar{\hat{r}}_\tau, \bar{\rho}_\tau)$ specifies the *distortion* of $\bar{\hat{r}}$ from $\bar{\rho}_\tau$ and is defined in the following form:

$$\Delta(\bar{\hat{r}}_\tau, \bar{\rho}_\tau) = \frac{1}{\tau + 1} \sum_{i=0}^{\tau} 1_{(r_i \neq \hat{r}_i)} \tag{9}$$

Parameter $\mathcal{K}$ specifies the *adversary knowledge* at time $t = \tau$ (which is the obfuscated track of node c in time interval $[0, \tau]$).

Assume that $\mathcal{S}_b(\tau)$ denotes a square centered at the centroid of the current obfuscation region (at time $t = \tau$) and has the edge length $b\alpha$. Subsequently, the privacy-level formula is rewritten as the following form:

$$\lambda(T, \bar{\rho}) = \int_{b=\frac{1}{3}}^{1} \sum_{\hat{r} \subset \mathcal{S}_b(\tau)} \mathbf{Pr}[\bar{r}_\tau = \bar{\hat{r}}_\tau | \mathcal{K}] \Delta(\bar{\hat{r}}_\tau, \bar{\rho}_\tau) \\ \times \frac{\mathrm{d}b}{1 - \frac{1}{3}} = \frac{3}{2} \int_{b=\frac{1}{3}}^{1} \sum_{\hat{r} \subset \mathcal{S}_b(\tau)} \mathbf{Pr}[\bar{r}_\tau = \bar{\hat{r}}_\tau | \mathcal{K}] \Delta(\bar{\hat{r}}_\tau, \bar{\rho}_\tau) \mathrm{d}b \tag{10}$$

In fact, parameter b specifies the value of boundary factor which is uniformly distributed in time interval $(1/3, 1)$.

A. Quantification of the Privacy Level

Before starting our discussion in this subsection, let's precisely define the following mobility model for node c: node c is at coordinates (x_0+x_t, y_0+y_t) at time t such that (x_0, y_0) represents the initial location of node c. Moreover, x_t and y_t specify two *scaled Wiener*[1] processes of variance σ^2.

Considering Algorithm 1, assume that function UPDATE is firstly called at time $t = T_1 + 1$. Here, we are calculating the privacy level in time interval $[0, \tau]$ such that τ is an integer not greater than T_1.

Assume that $\widehat{r_0} \subset \mathcal{S}_0$ (where $\mathcal{S}_0$ denotes the square of edge length $\alpha/3$ and centered at the centroid of the current obfuscated region) and $\widehat{r_i} \subset \mathcal{S}_b$ for every $i = 1, 2, \dots, \tau$. We can rewrite $\mathbf{Pr}(\bar{r} = \widehat{\bar{r}}|\mathcal{K})$ in the following form:

$$\begin{aligned}\mathbf{Pr}(\bar{r} = \widehat{\bar{r}}|\mathcal{K}) &= \mathbf{Pr}\big(\bigwedge_{i=0}^{\tau}(r_i = \widehat{r_i})|\mathcal{K}\big)\\ &= \mathbf{Pr}(r_0 = \widehat{r_0}) \times \mathbf{Pr}(r_1 = \widehat{r_1}|r_0 = \widehat{r_0}, \mathcal{K})\\ &\quad \times \mathbf{Pr}\big(r_2 = \widehat{r_2}|\bigwedge_{i=0}^{1}(r_i = \widehat{r_i}), \mathcal{K}\big)\\ &\quad \vdots\\ &\quad \times \mathbf{Pr}\big(r_\tau = \widehat{r_\tau}|\bigwedge_{i=0}^{\tau-1}(r_i = \widehat{r_i}), \mathcal{K}\big)\end{aligned} \tag{11}$$

Regarding Algorithm 1, it is easy to see that (As mentioned before, the initial location of node c always lies inside square $\mathcal{S}_0$):

$$p_0 = \mathbf{Pr}(r_0 = \widehat{r_0}|\mathcal{K}) = \frac{9}{\alpha^2} \tag{12}$$

Additionally, assuming $\widehat{r_i} = [\widehat{x_i}, \widehat{x_i}+1) \times [\widehat{y_i}, \widehat{y_i}+1)$, we obtain that ($i > 0$):

$$\begin{aligned}p_i &= \mathbf{Pr}\big(r_i = \widehat{r_i}|\bigwedge_{j=0}^{i-1}(r_j = \widehat{r_j}), \mathcal{K}\big)\\ &= \mathbf{Pr}\big((x_i, y_i) \in [\widehat{x_i}, \widehat{x_i}+1) \times [\widehat{y_i}, \widehat{y_i}+1)\\ &\qquad |\bigwedge_{j=0}^{i-1}(r_j = \widehat{r_j}), \mathcal{K}\big)\\ &= \mathbf{Pr}\big((x_i - x_{i-1}, y_i - y_{i-1}) \in [\widehat{x_i} - \widehat{x_{i-1}}, \widehat{x_i} -\\ &\qquad \widehat{x_{i-1}} + 1) \times [\widehat{y_i} - \widehat{y_{i-1}}, \widehat{y_i} - \widehat{y_{i-1}} + 1)\big)\\ &= \mathbf{Pr}\big(x_i - x_{i-1} \in [\widehat{x_i} - \widehat{x_{i-1}}, \widehat{x_i} - \widehat{x_{i-1}} + 1)\\ &\qquad \wedge y_i - y_{i-1} \in [\widehat{y_i} - \widehat{y_{i-1}}, \widehat{y_i} - \widehat{y_{i-1}} + 1)\big)\end{aligned} \tag{13}$$

Since x_t and y_t are Wiener processes of variance σ^2, we obtain the following equations ($\delta x_i = x_i - x_{i-1}$, $\delta y_i = y_i - y_{i-1}$, $\delta\widehat{x_i} = \widehat{x_i} - \widehat{x_{i-1}}$, and $\delta\widehat{y_i} = \widehat{y_i} - \widehat{y_{i-1}}$):

$$\begin{aligned}p_i &= \mathbf{Pr}\big(\delta\widehat{x_i} \leq \delta x_i < \delta\widehat{x_i} + 1\big)\\ &\quad \times \mathbf{Pr}\big(\delta\widehat{y_i} \leq \delta y_i < \delta\widehat{y_i} + 1\big)\end{aligned} \tag{14}$$

Regarding the definition of scaled Wiener process, $\delta x_i \sim N(0, \sigma^2)$ and $\delta y_i \sim N(0, \sigma^2)$. Consequently,

$$\begin{aligned}p_i &= \frac{1}{2}\big(\text{erf}(\frac{\delta\widehat{x_i}+1}{\sqrt{2\sigma^2}}) - \text{erf}(\frac{\delta\widehat{x_i}}{\sqrt{2\sigma^2}})\big)\\ &\quad \times \frac{1}{2}\big(\text{erf}(\frac{\delta\widehat{y_i}+1}{\sqrt{2\sigma^2}}) - \text{erf}(\frac{\delta\widehat{y_i}}{\sqrt{2\sigma^2}})\big)\end{aligned} \tag{15}$$

Considering that

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}\text{d}t, \tag{16}$$

we can simplify the value of p_i in the following form:

$$\begin{aligned}p_i &= \frac{1}{4} \times \big(\frac{2\sqrt{2\sigma^2}}{\sqrt{\pi}}\big)^2 \exp\big(-\frac{\delta\widehat{x_i}^2}{2\sigma^2}\big)\exp\big(-\frac{\delta\widehat{y_i}^2}{2\sigma^2}\big)\\ &= \frac{2\sigma^2}{\pi}\exp\big(-\frac{\delta\widehat{x_i}^2 + \delta\widehat{y_i}^2}{2\sigma^2}\big)\end{aligned} \tag{17}$$

Consequently, we obtain that:

$$\begin{aligned}\mathbf{Pr}(\bar{r} = \widehat{\bar{r}}|\mathcal{K}) &= \prod_{i=0}^{\tau} p_i\\ &= \frac{9(2\sigma^2)^\tau}{\alpha^2\pi^\tau}\exp\big(-\frac{\sum_{i=1}^{\tau}\delta\widehat{x_i}^2 + \delta\widehat{y_i}^2}{2\sigma^2}\big)\end{aligned} \tag{18}$$

which implies that:

$$\begin{aligned}\lambda(\tau, \bar{\rho}) =& \sum_{\widehat{\bar{r}}}\frac{9(2\sigma^2)^\tau\Delta(\widehat{\bar{r}}, \bar{\rho})}{\alpha^2\pi^\tau}\\ &\times \exp\big(-\frac{\sum_{i=1}^{\tau}\delta\widehat{x_i}^2 + \delta\widehat{y_i}^2}{2\sigma^2}\big)\end{aligned} \tag{19}$$

Finally, regarding Equation 10, we obtain a closed formula for the instantanious privacy-level of node c at time $t = \tau$:

$$\begin{aligned}\lambda(\tau, \bar{\rho}) =& \frac{27(2\sigma^2)^\tau}{2\alpha^2\pi^\tau}\int_{b=\frac{1}{3}}^{1}\sum_{\widehat{\bar{r}} \subset \mathcal{S}_b}\Delta(\widehat{\bar{r}}, \bar{\rho})\\ &\times \exp\big(-\frac{\sum_{i=1}^{\tau}\delta\widehat{x_i}^2 + \delta\widehat{y_i}^2}{2\sigma^2}\big)\text{d}b\end{aligned} \tag{20}$$

Equation 20 specifies the privacy-level for every time $\tau = 0, 1, \dots T_1$. Now, we are focusing on finding the privacy-level for $\tau > T_1$.

Let T_j denote the moment that Algorithm 1 calls function UPDATE for the j^{th} time. We claim that for every $\tau = T_j, T_j + 1, \dots, T_{j+1}$ and $j = 1, 2, \dots$, the privacy-level is obtained by the following equation:

$$\begin{aligned}\lambda(\tau, \bar{\rho}) =& \frac{27(2\sigma^2)^{\tau-T_j}}{2\alpha^2\pi^{\tau-T_j}}\int_{b=\frac{1}{3}}^{1}\sum_{\widehat{\bar{r}}_j \subset \mathcal{S}_b}\Delta(\widehat{\bar{r}}_j, \bar{\rho}_j)\\ &\times \exp\big(-\frac{\sum_{i=1}^{\tau-T_j}\delta\widehat{x_{i+T_j}}^2 + \delta\widehat{y_{i+T_j}}^2}{2\sigma^2}\big)\text{d}b\end{aligned} \tag{21}$$

where $\bar{\rho} = \rho_0, \rho_1, \dots, \rho_\tau$, $\bar{\rho}_j = \rho_{T_j}, \rho_{T_j+1}, \dots, \rho_\tau$, and $\widehat{\bar{r}}_j = \widehat{r_{T_j}}, \widehat{r_{T_j}}, \dots, \widehat{r_\tau}$. The reason is that after the j^{th} update, the location of node c (r_{T_j}) is uniformly distributed over the area inside the square of edge length $\alpha/3$ and centered

[1] W_t is called a Wiener process if $W_0 = 0$, function $t \mapsto W_t$ is almost surely everywhere continuous, and W_t has independent increment with $W_t - W_s \sim N(0, t-s)$ for every $0 \leq s < t$. W'_t is a scaled Wiener process of variance σ^2, if W'_t/σ specifies a Wiener process.

at the centroid of new obfuscated region. This means that the probability distribution of process r_t is exactly repeated in moments $0, T_1, T_2, \ldots$. Additionally, as Wiener process is stationary, the calculation of instantaneous privacy-level in time interval $[T_j, T_{j+1}]$ is the same as what we did for interval $[0, T_1]$. Note that the privacy-level at boundary times $\tau = T_1, T_2, \ldots$ can be computed using two different formula. The reason is that the privacy-level will change before and after calling the UPDATE function.

B. Trade-Off between Privacy-Level and Error-Tolerance

Every location privacy-preserving mechanism enables each mobile node to specify the level of privacy that it desires and the maximum spatial error tolerance that it is willing to accept when sending queries to the LBSs. As mentioned before, there exists a trade-off between the privacy-level and maximum error tolerance which is represented by ε. In this subsection, we find a relation between these two values for the clients that use our proposed LPPM.

We define the error-tolerance as the *expected value of the (Euclidean) distance between node c's actual location (r_t) and the LBSs' guess (g_t)* such that g_t specifies the region guessed by the LBS. As we are only concerned about the maximum error-tolerance, we skip the calculation of the expected value. Instead, we find an upper-bound for the distance between g_t and r_t. Since the LBS is only aware of the obfuscated region which is a square of edge length α, we obtain that:

$$|g_t - r_t| \leq \sqrt{2}\alpha \qquad \forall t \tag{22}$$

such that $\sqrt{2}\alpha$ is the diameter of the obfuscated square. As the result,

$$\varepsilon(t) \leq \sqrt{2}\alpha \tag{23}$$

Additionally, it is easy to show that regarding Equation 8, privacy-level at time $t = 0$ is equal to $1 - 9/\alpha^2$ or $1 - \lambda(0, \bar{\rho}) = O(\varepsilon^{-2}(t))$. This means that if node c accepts higher spatial error, the initial privacy-level will increase significantly. Later in Section 5, we use Equations 21 and 23 to show this trade-off in more general form.

V. Simulation Results

We implement our proposed LPPM using MATLAB R2013a to evaluate its performance in practice. Figure 2 illustrates how the LPPM hides the instantaneous location and path movement of a randomly walking node. In order to find the value of instantaneous privacy level, we use Equation 21. In this specific case, $T_1 = 162$, $T_2 = 179$. Figures 3 and 4 respectively show the plot of instantaneous privacy level in time intervals $[0, 161]$ and $[162, 178]$. As you see, the value of privacy level becomes very close to one at the first few time units. Additionally, after every update, the instantaneous privacy level drops to $1 - 9/\alpha^2 = 0.9955$; i.e the minimum privacy level is $1 - 9/\alpha^2 = 1 - \Theta(\varepsilon^{-2})$ (trade-off between λ and ε). Figure 5 depicts the plot of function $\log(1 - \lambda)$ over time interval $[162, 178]$. As you see, the privacy level gets close to one with *exponential* rate (as $\log(1 - \lambda)$ linearly grows).

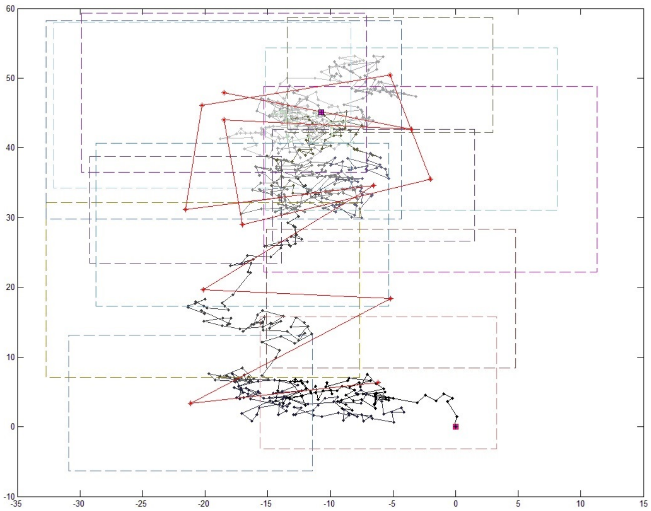

Fig. 2: Track of node c walking randomly on the Euclidean plane (x-y Cartesian coordinates) in time interval $[0, 700]$. In this case, $\alpha = 45$, $\sigma^2 = 1$, and $(x_0, y_0) = (0, 0)$. As you see, the track of node c has been shown with black/gray color (the darker, the earlier). Additionally, the start and end point of the track has been specified with asterisk and plus sign respectively. Dashed squares show the boundary of obfuscated region at different times (edge length of each square is $\mathcal{B}\alpha$ where $\mathcal{B}$ is the boundary factor). The red track specifies the obfuscated track which connects the centroids of subsequent squares.

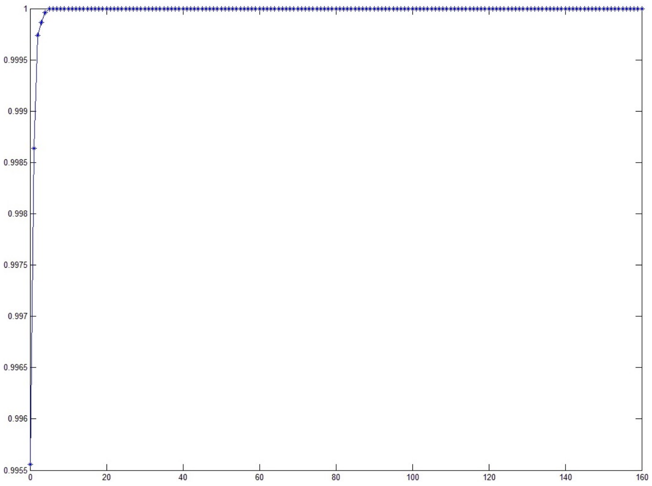

Fig. 3: Plot of instantaneous privacy-level over time interval $[0, 161]$ (before the first update). y axis specifies $\lambda(t, \bar{\rho})$ while $x = t$.

VI. Summary and Conclusion

In this paper, we proposed a novel LPPM based on location obfuscation for protecting the location privacy of a randomly walking node on the Euclidean plane which continuously exposes its location to an LBS (or possibly an adversary). Then, we quantified the privacy-level of the node over time by computing the expected distortion of adversary's guess from the reality of node's movement path (track). Additionally,

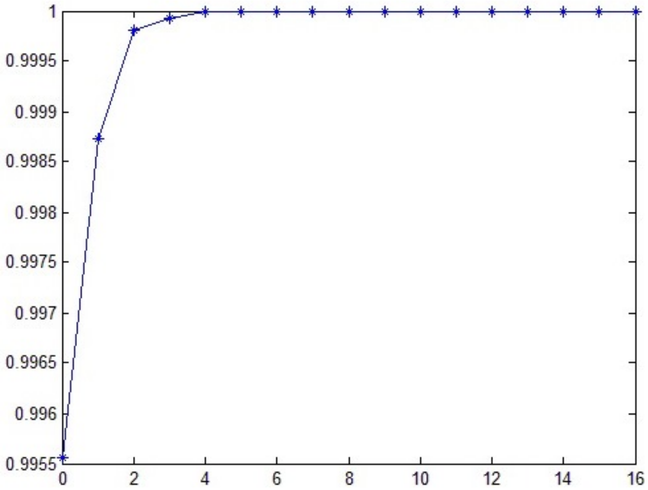

Fig. 4: Plot of instantaneous privacy-level over time interval $[162, 178]$ (after the first update and before the second one). y axis specifies $\lambda(t, \bar{\rho})$ while $x = t - 162$.

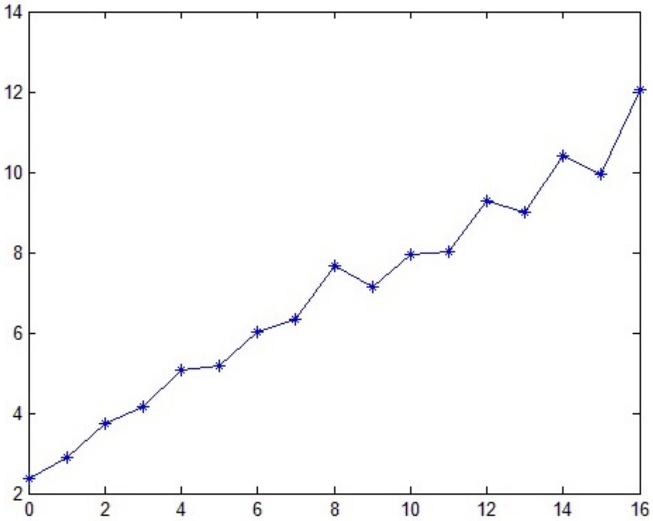

Fig. 5: Plot of function $\log_{10}(1 - \lambda)$ over time interval $[162, 178]$. y axis specifies $\log_{10}(1-\lambda(t, \bar{\rho}))$ while $x = t-162$.

the trade-off between the privacy level and maximum error tolerance of the walking node was examined closely. As the result, the minimum privacy level occurs at the moments that the obfuscated location is being updated and we obtained that $1 - \lambda_{\min} = \Theta(\varepsilon_{\max}^{-2})$. Finally, we used some simulations to illustrate our theoretical results and prove the efficacy of our method in practice.

Acknowledgment

The authors would like to thank...

References

[1] Rinku Dewri, “Location Privacy and Attacker Knowledge: Who Are We Fighting Against?,” *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, Volume 96, pp 96-115, 2012.

[2] Ming Li, Sergio Salinas, Arun Thapa, Pan Li, “n-CD: A Geometric Approach to Preserving Location Privacy in Location-Based Services,” *Proc. IEEE INFOCOM*, 2013.

[3] Rui Shi, Mayank Goswami, Jie Gao, and Xianfeng Gu, “Is Random Walk Truly Memory-less - Traffic Analysis and Source Location Privacy under Random Walks,” *INFOCOM, 2013 Proceedings IEEE*, Pages 3021-3029, Turin, Italy, 2013.

[4] M. Gruteser and D. Grunwald, “Anonymous usage of location-based services through spatial and temporal cloaking,” *in ACM Mobisys’03*, May 2003.

[5] B. Gedik and L. Liu, ”Protecting location privacy with personalized k-anonymity: Architecture and algorithms,” *IEEE Transactions on Mobile Computing*, vol. 7, no. 1, pp. 118, January 2008.

[6] J. Meyerowitz and R. R. Choudhury, “Hiding stars with fireworks: Location privacy through camouflage,” *in Proceedings of ACM MobiCom*, Beijing, China, September 2009.

[7] M. F. Mokbel, C. Y. Chow, and W. G. Aref, “The new casper: Query processing for location services without compromising privacy,” *in Proceedings of VLDB*, 2006.

[8] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias, “Preventing location-based identity inference in anonymous spatial queries,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 12, pp. 1719 1733, December 2007.

[9] B. Gedik and L. Liu, “Location privacy in mobile systems: A personalized anonymization model,” *in Proceedings of IEEE ICDCS*, Columbus, Ohio, June 2005.

[10] Chi-Yin Chow, Mohamed F. Mokbel, Xuan Liu, “A peer-to-peer spatial cloaking algorithm for anonymous location-based service,” *in Proceedings of ACM GIS*, Arlington, Virginia, November 2006.

[11] A. Beresford and F. Stajano, “Location privacy in pervasive computing,” *IEEE Pervasive Computing*, vol. 2, no. 1, pp. 4655, 2003.

[12] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady, “Preserving privacy in gps traces via uncertainty-aware path cloaking,” *in Proceedings of ACM CCS 2007*, Alexandria, VA, US, January 2007.

[13] H. Kido, Y. Yanagisawa, and T. Satoh, “An anonymous communication technique using dummies for location-based services,” *in Proceedings of IEEE ICPS*, Santorini, Greece, July 2006.

[14] H. Lu, C. S. Jensen, and M. L. Yiu, “Pad: privacy-area aware, dummy-based location privacy in mobile services,” *in Proceedings of ACM MobiDE*, Vancouver, Canada, June 2008.

[15] M. Duckham and L. Kulik, “A formal model of obfuscation and negotiation for location privacy,” *in Proceedings of International Conference on Pervasive Computing*, Munich, Germany, May 2005.

[16] C. A. Ardagna, M. Cremonini, S. D. C. di Vimercati, and P. Samarati, “An obfuscation-based approach for protecting location privacy,” *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 1, pp. 1327, January 2011.

[17] A. Pingley, W. Yu, N. Zhang, X. Fu, and W. Zhao, “Cap: A context-aware privacy protection system for location-based services,” *in Proceedings of IEEE ICDCS*, Montreal, Canada, June 2009.

[18] M. Damiani, E. Bertino, and C. Silvestri, “Probe: An obfuscation system for the protection of sensitive location information in lbs,” *Technical Report 2001-145*, CERIAS, 2008.

[19] Zhenqiang Gong, Guang-Zhong Sun, and Xing Xie, “Protecting Privacy in Location-based Services Using K-anonymity without Cloaked Region,” *Eleventh International Conference on Mobile Data Management*, Kansas City, MO, USA, 2010.

[20] Reza Shokri, George Theodorakopoulos, George Danezis, Jean-Pierre Hubaux, and Jean-Yves Le Boudec, “Quantifying Location Privacy: The Case of Sporadic Location Exposure,” *PETS 2011*, LNCS 6794, pp. 5776, 2011.

[21] Reza Shokri, George Theodorakopoulos , Carmela Troncoso, Jean-Pierre Hubaux, and Jean-Yves Le Boudec, “Protecting Location Privacy: Optimal Strategy against Localization Attacks,” *CCS ’12 Proceedings of the 2012 ACM conference on Computer and communications security*, Pages 617-627, New York, NY, USA, 2012.

[22] Reza Shokri, George Theodorakopoulos, Jean-Yves Le Boudec, and Jean-Pierre Hubaux, “Quantifying Location Privacy,” *Security and Privacy (SP), 2011 IEEE Symposium on*, Pages 247 - 262, Berkeley, CA, May 2011.