Quantifying Location Privacy: The Case of Sporadic Location Exposure

[†]Reza Shokri, [†]George Theodorakopoulos, [‡]George Danezis,
[†]Jean-Pierre Hubaux, and [†]Jean-Yves Le Boudec

[†] LCA, EPFL, Lausanne, Switzerland, [‡] Microsoft Research, Cambridge, UK
[†] firstname.lastname@epfl.ch, [‡] gdane@microsoft.com

Abstract. Mobile users expose their location to potentially untrusted entities by using location-based services. Based on the frequency of location exposure in these applications, we divide them into two main types: *Continuous* and *Sporadic*. These two location exposure types lead to different threats. For example, in the continuous case, the adversary can track users over time and space, whereas in the sporadic case, his focus is more on localizing users at certain points in time. We propose a systematic way to quantify users' location privacy by modeling both the location-based applications and the location-privacy preserving mechanisms (LPPMs), and by considering a well-defined adversary model. This framework enables us to customize the LPPMs to the employed location-based application, in order to provide higher location privacy for the users. In this paper, we formalize *localization* attacks for the case of sporadic location exposure, using Bayesian inference for Hidden Markov Processes. We also quantify user location privacy with respect to the adversaries with two different forms of background knowledge: Those who only know the geographical distribution of users over the considered regions, and those who also know how users move between the regions (i.e., their mobility pattern). Using the Location-Privacy Meter tool, we examine the effectiveness of the following techniques in increasing the expected error of the adversary in the localization attack: Location obfuscation and fake location injection mechanisms for anonymous traces.

1 Introduction

Mobile devices equipped with various positioning systems have paved the way for the emergence of numerous interesting location-based services. Unfortunately, this phenomenon has opened the door to many new threats to users' privacy, as untrusted entities (including the service providers themselves) can track users' locations and activities over time by observing their location-based queries.

Location-based applications, in effect, expose over time some of the locations of users to curious observers (adversaries) who might collect this information for various monetary or malicious purposes. In most of such applications, users share/expose their location in a *sporadic* manner as opposed to a *continuous* manner. Widely used location-based services (LBSs), such as local search applications for finding nearby points-of-interests or nearby friends, are good examples of this type of applications.

To protect users' location privacy, location-privacy preserving mechanisms (LPPMs) can be used as a filter between the location-based applications and the potentially adversarial observers. Many interesting LPPMs have been proposed for sporadic applications. Anonymization and obfuscation of users' location events (e.g., LBS queries) are the most popular techniques.

However, so far there is no theoretical framework to both formalize the effectiveness of various location-privacy preserving mechanisms, and to take into account the characteristics of the underlying location-based application. To fill this gap, we leverage on the framework that we have proposed and used in our previous contributions [17–19]. More specifically, in this paper we make three major contributions. First, we formalize the location exposure in location-based services, particularly their location-exposure pattern, and add it to the framework. Second, we build upon this formalization to quantitatively evaluate the effectiveness of various LPPMs, notably the *fake-location injection* as a mechanism to protect location privacy of users. Third, we provide an analytical model, based on Hidden Markov Processes, for localization attacks. We extend the Location-Privacy Meter tool [1] to support these new features. We use the incorrectness of the adversary (i.e., his expected estimation error) [19] in localizing users over time as the location-privacy metric. We also implement some example location-based applications in our evaluation tool and assess the effectiveness of various LPPMs.

It is noteworthy that we do not address the problem of quality-of-service degradation in location-based services due to the usage of a location-privacy preserving mechanism. This issue is orthogonal to our objective in this paper, which is to provide methods to accurately assess the loss of location privacy.

The rest of the paper is organized as follows. In Section 2, we describe our framework. In Section 3, we detail the localization attack, based on Bayesian analysis. In Section 4, we evaluate the approach on a concrete example. We provide the related work in Section 5, and conclude the paper in Section 6.

2 Framework

2.1 Mobile Users

We consider $\mathcal{U} = \{u_1, u_2, \ldots, u_N\}$ a set of N mobile users who move within an area that is partitioned into M distinct regions (locations) $\mathcal{R} = \{r_1, r_2, \ldots, r_M\}$. Time is considered to be discrete, and the set of time instants when the location of users may be observed is $\mathcal{T} = \{1, \ldots, T\}$. The precision at which we want to represent the user mobility determines the granularity of the space and time. For example, regions can be of a city/block size, and two successive time instants can be a day/hour apart, if the mobility is supposed to have a low/high precision.

The spatiotemporal position of users is modeled through events and traces. An *event* is defined as a triplet $\langle u, r, t \rangle$, where $u \in \mathcal{U}$, $r \in \mathcal{R}$, $t \in \mathcal{T}$. A *trace* of user u is a T-size vector of events $a_u = (a_u(1), a_u(2), \ldots, a_u(T))$. The set of all traces that may belong to user u is denoted by $\mathcal{A}_u$. Notice that, of all the traces in $\mathcal{A}_u$, exactly one is the true trace that user u created in the time period of interest ($t = 1 \ldots T$); the true trace, denoted by a_u, is called the *actual trace* of user u, and its events are called the *actual events* of user u. The set of all possible traces of all users is denoted by $\mathcal{A} = \mathcal{A}_{u_1} \times \mathcal{A}_{u_2} \times \ldots \times \mathcal{A}_{u_N}$; the member of $\mathcal{A}$ that is actually created by the N users is denoted by a and it is equal to $(a_{u_1}, a_{u_2}, \ldots, a_{u_N})$. The vector of actual traces a is in fact a sample from the random variable $\boldsymbol{A}$ that is distributed according to $p(\cdot) = \Pr\{\boldsymbol{A} = \cdot\}$. The distribution p reflects the joint mobility pattern of the users. We refer to each marginal distribution p_u as the *mobility profile* of user u, that is $a_u \sim p_u(\cdot) = \Pr\{\boldsymbol{A_u} = \cdot\}$.

In this paper, we assume that the users' profiles are independent of each other, i.e., $p(\cdot) = \prod_u p_u(\cdot)$. In other words, the location of a user is independent of others, *given* the user's profile (i.e., there is a conditional independence between the users' locations). As users tend to have different mobility patterns at different *time periods* (e.g., morning vs. afternoon, or weekday vs. weekend), we assume the users' profiles to be time-period dependent. The set of time instants in $\mathcal{T}$ is partitioned by the time periods. Notice that the independence of user profiles means that we are ignoring social correlations among users, e.g., we ignore information about friendships among users; this is outside the scope of this paper. However, because of the time dependence, we do take into account indirect correlation among the users' locations, for instance traffic jams in the morning and in the evening.

Further, we assume that the mobility of a user is modeled as a Markov chain on the set of regions. So, for user u, the distribution p_u of actual traces can be computed using the transition matrix of its Markov chain. Each state of the Markov chain represents a region and a time period. We use $p_u^\tau(r, s)$ to indicate the probability of a transition from region r to s by user u in time period τ. We also use $\pi_u^\tau(r)$ to indicate the probability that user u is in region r in time period τ, according to the stationary probability distribution of p_u^τ.

Thus, we illustrate the mobility profile of users using a first-order Markov chain model which is dependent on time (periods). It is worth noting that the Markov chain model can be turned into a more powerful (yet more complex) model depending on how the states of the chain are defined. If states represent complex previous location behaviors (past n location, or locations in past day), then the model can become arbitrarily accurate.

2.2 Location-Based Applications

We differentiate among the location-based applications according to the frequency at which the users' locations are exposed. On one end of the spectrum, users' locations are continuously exposed through the application, whereas on the other end, there are applications using which users expose their location in a rather sporadic manner. In a nutshell, an application is considered to be *sporadic* if the exposed locations from the users are sparsely distributed over time, and it is considered *continuous* otherwise.

In this paper, we focus on the sporadic case (for some examples of the continuous case see [9, 11]). Examples for this type of systems are (i) location-based services where users make location-stamped queries concerning their nearby points of interest in order to receive contextual information, and (ii) location-sharing applications by which users can share their location with their friends, or with people-centric sensing servers, e.g., when they report about a social event.

Let $x_u \in \{0,1\}^T$ be a vector that shows which actual events of user u are exposed through the application. In effect, x_u acts as a bit-mask, for example, if $x_u(t) = 1$, then $a_u(t)$ is exposed.

We define a location-based application as a function that maps actual traces $a \in \mathcal{A}$ to a random variable $\boldsymbol{X}$ that takes values in the set $\mathcal{X} = \{0,1\}^{N \times T}$. The corresponding probability distribution function $\Pr\{\boldsymbol{X} = x | \boldsymbol{A} = a, p\}$ can be computed as follows, considering that mobile users usually make use of the location-based applications independently at each time instant:

$$\Pr\{\boldsymbol{X} = x | \boldsymbol{A} = a, p\} = \prod_u \prod_t \Pr\{\boldsymbol{X_u(t)} = x_u(t) | \boldsymbol{A_u(t)} = a_u(t), p\} \quad (1)$$

where p is the set of all users' actual mobility profiles.

2.3 Location-Privacy Preserving Mechanisms

The service provider, or any other entity that can access to the users' locations through some location-based applications, is considered as the adversary (or the observer) in this paper. Such an entity can indeed de-anonymize the users' traces and eventually localize users over time by relying on its background knowledge about users (e.g., their home/work address, their mobility patterns). We denote the background knowledge of the adversary about users by $\mathcal{K}$.

In order to thwart such threats, the users distort their exposed locations before an untrusted entity can see them. Location-privacy preserving mechanisms (LPPMs) are put in place to perform this distortion. LPPMs can be implemented both in a centralized architecture, by means of a trusted third party, and in a distributed architecture, i.e., an independent installation on each of the mobile devices. We abstract away these details and provide a generic model: the LPPMs act on the set of exposed traces and produce a set of traces that are observed by the untrusted entities. The LPPM is assumed to modify the set of exposed events using anonymization and obfuscation techniques. We now describe each of these in turn.

In the anonymization process, the username part of each trace is replaced by a *user pseudonym* in the set $\mathcal{U}' = \{1, ..., N\}$. The *anonymization* mechanism that we consider is the random permutation. That is, a permutation of the users is chosen uniformly at random among all $N!$ permutations and each user's pseudonym is her position in the permutation. More formally, the anonymization mechanism selects, independent of everything else, a permutation σ according to the probability distribution function $\Pr\{\boldsymbol{\Sigma} = \sigma\} = \frac{1}{N!}$, and each user's pseudonym is $\sigma(u) \in \mathcal{U}'$.

Notice that the pseudonym of a user remains the same for the whole time period $t = 1, \ldots, T$. The larger the value of T, the easier it is, in general, for the adversary to de-anonymize the users. In this paper, we do not study changing pseudonyms. However, we do study the effect of T on the privacy, and in particular the anonymity, of users (Section 4). Knowing the relation between T and user privacy is useful for deciding when to change a pseudonym, for example, when user privacy drops below a certain threshold.

In the obfuscation process, three event/trace transformations can happen:

- The location part of each exposed event can be replaced by a *location pseudonym* in the set $\mathcal{R}' = \mathcal{P}(R) = \{r'_1, ..., r'_{2^M}\}$. Each location pseudonym corresponds to a subset of regions in $\mathcal{R}$. Notice that each region can be obfuscated to a different location pseudonym each time it is encountered in a trace, whereas each user is always anonymized to the same user pseudonym.[1]
- Fake location-pseudonyms can be injected at times that the user does not expose anything (it is equivalent to say that the LPPM selects a fake location and then obfuscates it).
- Some of the exposed events can be removed (become hidden).

The LPPM, as the combination of the two processes, probabilistically maps exposed traces $(a, x) \in \mathcal{A} \times \mathcal{X}$ to obfuscated and anonymized traces. The output is a random variable $\boldsymbol{O}$ that takes values in the set $\mathcal{O}$, which is the set of all possible obfuscated and anonymized traces of all users. Such a trace is composed of T events of the form $o_{u'}(t) = \langle u', r', t \rangle$, where $u' \in \mathcal{U}'$, $r' \in \mathcal{R}'$, for $t = \{1, 2, \cdots, T\}$. A complete trace is denoted by $o_{u'}$.

In this paper, we study the case where each exposed event of a user is obfuscated independently of other events which belong to that user or other users. The mobility profiles of all users are used by the LPPM in the process of obfuscating users' locations. This knowledge of the users' profiles enables us to design strong LPPMs against the adversary who also relies on this type of information. The probability of a given output o is then computed as follows:

$$\begin{aligned} &\Pr\{\boldsymbol{O} = o | \boldsymbol{X} = x, \boldsymbol{A} = a, p\} = \\ &= \sum_{\sigma} \prod_{u'} \prod_{t} \underbrace{\Pr\{\boldsymbol{O}_{u'}(t) = o_{u'}(t) | \boldsymbol{\Sigma} = \sigma, \boldsymbol{X} = x, \boldsymbol{A} = a, p\}}_{\text{Obfuscation mechanism}} \\ &\qquad\qquad \cdot \underbrace{\Pr\{\boldsymbol{\Sigma} = \sigma | \boldsymbol{X} = x, \boldsymbol{A} = a, p\}}_{\text{Anonymization mechanism}} \end{aligned} \tag{2}$$

Notice that, in general, employing an LPPM reduces the quality of the information provided to the location-based service. Consequently, the quality of service that the user receives is also reduced. Therefore, there exists a tradeoff between the effectiveness of the LPPM and the quality of service for the user. Addressing this tradeoff is beyond the scope of this paper. Our objective is to evaluate the privacy that a given LPPM provides to the users.

[1] In this paper, we do not consider the obfuscation of the events' time-stamps, and leave it for future work.

2.4 Attacker

The adversary observes o, and by relying on his background knowledge $\mathcal{K}$, tries to infer the actual location of users. The adversary is assumed to be aware of the type and the characteristics of the location-based application, and also the location-privacy preserving mechanism. In order to infer the location of users, the adversary has to reverse the two mechanisms. The adversary's ultimate goal is then formally defined as calculating the following probability distribution function:

$$h_o(\hat{a}) = \Pr\{\boldsymbol{A} = \hat{a} | \boldsymbol{O} = o, \mathcal{K}\} \tag{3}$$

2.5 Location-Privacy Metric

We quantify the location privacy of users as the error of the adversary in estimating the actual location of users. The metric is justified in [19], and its superiority to other metrics, such as k-anonymity and entropy, is shown qualitatively and quantitatively. According to the expected-estimation-error metric, the users' location privacy is computed as follows:

$$\mathrm{LP} = \sum_{\hat{a} \in \mathcal{A}} h_o(\hat{a}) \Delta(a, \hat{a}) \tag{4}$$

where $\Delta(a, \hat{a})$ is a distortion function that determines the distance between actual traces a and hypothesized traces $\hat{a}$. In this paper, we use the following distortion function:

$$\Delta(a, \hat{a}) = \frac{1}{N \cdot T} \sum_u \sum_t 1_{a_u(t) \neq \hat{a}_u(t)} \tag{5}$$

which makes LP the average probability of error of the adversary in estimating the actual location of users over time. Note that, the location privacy of each user can be computed separately in the same way.

3 Localization Attack

We define the goal of the adversary to be the localization of users over time: That is, for a given user at a given time instant, the adversary computes the probability distribution over regions where the user might be at that specific time instant, considering the observed traces. More formally, the adversary computes $\Pr\{\boldsymbol{A_u(t)} = \langle u, t, r\rangle | o, \mathcal{K}\}$ for user u at time instant t for all regions $r \in \mathcal{R}$. We call this the *localization attack*.

As an aside, more general objectives can be imagined for the attacker. The most general one is to recover all traces of all users, i.e., to compute the probability $\Pr\{\boldsymbol{A} = \cdot | \boldsymbol{O} = o, \mathcal{K}\}$ as in (3).

Monte Carlo methods can be used to compute any desired probability in our framework. At its base, a Monte Carlo method uses repeated sampling from an appropriate distribution to estimate the desired probability. In our case, sampling from the distribution that is appropriate for the most general objective, $\Pr\{\boldsymbol{A} = \cdot | \boldsymbol{O} = o, \mathcal{K}\}$, is computationally inefficient for large user populations and long time intervals. Even for the localization attack, the space from which the Monte Carlo method needs to sample includes all the $N!$ permutations of user-pseudonym assignments. Therefore, we choose a different method, which can be applied more generally.

We split the localization attack into two parts: de-anonymization, and de-obfuscation. In the first step, we find the most likely assignments between users and pseudonyms. Formally, we compute

$$\sigma^* = \underbrace{\arg\max_\sigma \Pr\{\boldsymbol{\Sigma} = \sigma | o, \mathcal{K}\}}_{\text{de-anonymization}}. \tag{6}$$

Then, given this assignment, we compute the probability distribution of the given user's location at the given time instant.

$$\Pr\{\boldsymbol{A_u(t)} = \langle u, t, r \rangle | o, \mathcal{K}\} \approx \underbrace{\Pr\{\boldsymbol{A_u(t)} = \langle u, t, r \rangle | \boldsymbol{\Sigma} = \sigma^*, o, \mathcal{K}\}}_{\text{de-obfuscation}} \tag{7}$$

We use Bayesian inference in order to perform both the de-anonymization and the de-obfuscation. Both steps have polynomial-time complexity (in N and T), so they are computationally efficient even for large problem sizes.

Notice that this computation is an approximation of the a-posteriori probability $\Pr\{\boldsymbol{A_u(t)} = \langle u, t, r \rangle | o, \mathcal{K}\}$, which can be written as a weighted sum as follows (we omit, but still imply the existence of, $\mathcal{K}$):

$$\begin{aligned} \Pr\{\boldsymbol{A_u(t)} = \langle u, t, r \rangle | o\} &= \sum_\sigma \Pr\{\boldsymbol{A_u(t)} = \langle u, t, r \rangle, \sigma | o\} \\ &= \sum_\sigma \Pr\{\boldsymbol{A_u(t)} = \langle u, t, r \rangle | \sigma, o\} \Pr\{\sigma | o\} \end{aligned} \tag{8}$$

In effect, our approximation replaces the weighted sum with the probability $\Pr\{\boldsymbol{A_u(t)} = \langle u, t, r \rangle | \sigma^*, o\}$. We call this the *zeroth-order* approximation.

Our approximation can be made arbitrarily precise, at the cost of extra computations, in the following way. The basic idea is to separate the permutations, over which the summation is done, into N groups according to the pseudonym that they assign to user u (group 1 assigns pseudonym u'_1 to user u, group 2 assigns pseudonym u'_2, etc.). Without loss of generality, we assume that u is u_1.

$$
\begin{aligned}
&\Pr\left\{\boldsymbol{A_u(t)} = \langle u,t,r\rangle | o\right\} = \sum_{\sigma} \Pr\left\{\boldsymbol{A_u(t)} = \langle u,t,r\rangle, \sigma | o\right\} = \\
&= \sum_{u_1' \in \mathcal{U}'} \quad \sum_{\sigma:\sigma(u_1)=u_1'} \Pr\left\{\boldsymbol{A_u(t)} = \langle u,t,r\rangle, \sigma | o\right\} \\
&= \sum_{u_1' \in \mathcal{U}'} \quad \sum_{\sigma:\sigma(u_1)=u_1'} \Pr\left\{\boldsymbol{A_u(t)} = \langle u,t,r\rangle | \sigma(u_1) = u_1', o_{u_1'}\right\} \Pr\left\{\sigma | o\right\} \\
&= \sum_{u_1' \in \mathcal{U}'} \left(\Pr\left\{\boldsymbol{A_u(t)} = \langle u,t,r\rangle | \sigma(u_1) = u_1', o_{u_1'}\right\} \sum_{\sigma:\sigma(u_1)=u_1'} \Pr\left\{\sigma | o\right\} \right) \qquad (9)
\end{aligned}
$$

It is computationally infeasible to compute the second sum explicitly. So, we can do the *first-order* approximation: we replace the sum with the maximum of the quantity $\Pr\{\mathit{\Sigma} = \sigma | o\}$ over all indicated permutations $\sigma : \sigma(u_1) = u_1'$. That is, for each $u_1' \in \mathcal{U}'$ we compute the maximum $\Pr\{\mathit{\Sigma} = \sigma | o\}$ over all permutations that assign the pseudonym u_1' to user u_1. Then, in the first sum, we use this maximum as the weight for the probability $\Pr\left\{\boldsymbol{A_u(t)} = \langle u,t,r\rangle | \sigma(u_1) = u_1', o_{u_1'}\right\}$. Finding the maximum is a Maximum Assignment Problem, which is solvable in polynomial time; we need to find N such maxima, one for each value of $u_1' \in \mathcal{U}'$. Therefore, the whole computation is still polynomial, although longer than our original approximation.

However, the successive approximation need not stop at the first order. Instead of computing the maximum $\Pr\{\boldsymbol{\Sigma} = \sigma | o\}$ over all permutations that assign the pseudonym u_1' to user u_1, we can expand the second sum as follows:

$$
\sum_{\sigma:\sigma(u_1)=u_1'} \Pr\left\{\boldsymbol{\Sigma} = \sigma | o\right\} = \sum_{u_2' \in \mathcal{U}' \setminus \{u_1'\}} \quad \sum_{\substack{\sigma:\sigma(u_1)=u_1', \\ \sigma(u_2)=u_2'}} \Pr\left\{\boldsymbol{\Sigma} = \sigma | o\right\} \qquad (10)
$$

Now, as before, we can approximate the second sum by a maximum over the indicated permutations, and use the computed maxima (one for each value of u_2') as weights to compute the weighted sum. Alternatively, we can keep improving the approximation by considering user u_3, and so on. If we do this for all users, then we will have computed the exact value of $\Pr\{\boldsymbol{\Sigma} = \sigma | o\}$. In this paper, we stay at the zeroth-order approximation, as it is shown in (6) and (7).

De-anonymization: In order to obtain the σ^* of (6), we need to maximize the probability

$$
\begin{aligned}
&\Pr\left\{\boldsymbol{\Sigma} = \sigma | o, \mathcal{K}\right\} = \Pr\left\{o | \boldsymbol{\Sigma} = \sigma, \mathcal{K}\right\} \cdot \underbrace{\frac{\Pr\left\{\boldsymbol{\Sigma} = \sigma | \mathcal{K}\right\} \equiv \frac{1}{N!}}{\Pr\left\{o | \mathcal{K}\right\}}}_{\text{constant}}, \\
&\text{where } \Pr\left\{o | \boldsymbol{\Sigma} = \sigma, \mathcal{K}\right\} = \prod_{u'} \Pr\left\{o_{u'} | \boldsymbol{\Sigma} = \sigma, \mathcal{K}\right\}. \qquad (11)
\end{aligned}
$$

Thus, $\sigma^* = \arg\max_\sigma \Pr\{\boldsymbol{\Sigma} = \sigma | o, \mathcal{K}\} = \arg\max_\sigma \prod_{u'} \Pr\{o_{u'} | \boldsymbol{\Sigma} = \sigma, \mathcal{K}\}$. Notice that, given the assignment of a user u to the pseudonym u', the probability $\Pr\{o_{u'} | \boldsymbol{\Sigma} = \sigma, \mathcal{K}\}$ is independent of all other user-pseudonym assignments. So, to find the most likely assignment σ^*, we first compute $\Pr\{o_{u'} | \sigma(u) = u', \mathcal{K}\}$ for all pairs of $u \in \mathcal{U}$ and $u' \in \mathcal{U}'$. Then, we construct a complete weighted bipartite graph whose disjoint sets of vertices are $\mathcal{U}$ and $\mathcal{U}'$ and the weight on the edge between given vertices u and u' is the likelihood $\Pr\{o_{u'} | \sigma(u) = u', \mathcal{K}\}$. In order to obtain σ^*, we then solve the maximum weight assignment problem for this graph (see also [21]). In our simulation, we use the Hungarian algorithm in order to solve this problem, which is a special case of a linear program.

De-obfuscation: Given the most likely user-pseudonym assignment σ^*, we perform the de-obfuscation (7) as follows:

$$\begin{aligned} &\Pr\{\boldsymbol{A_u(t)} = \langle u, t, r\rangle | \boldsymbol{\Sigma} = \sigma^*, o, \mathcal{K}\} = \\ &= \Pr\{\boldsymbol{A_u(t)} = \langle u, t, r\rangle | o_{u'}, \sigma^*(u) = u', \mathcal{K}\} \\ &= \frac{\Pr\{\boldsymbol{A_u(t)} = \langle u, t, r\rangle, o_{u'} | \sigma^*(u) = u', \mathcal{K}\}}{\sum_{s \in \mathcal{R}} \Pr\{\boldsymbol{A_u(t)} = \langle u, t, s\rangle, o_{u'} | \sigma^*(u) = u', \mathcal{K}\}} \end{aligned} \tag{12}$$

The distribution over all regions r is obtained by computing the probability $\Pr\{\boldsymbol{A_u(t)} = \langle u, t, r\rangle, o_{u'} | \sigma^*(u) = u', \mathcal{K}\}$ for all $r \in \mathcal{R}$.

Adversary Knowledge: The de-anonymization and the de-obfuscation processes have been reduced, as seen in (11) and (12), to the computation of the probabilities $\Pr\{o_{u'} | \sigma(u) = u', \mathcal{K}\}$ and $\Pr\{\boldsymbol{A_u(t)} = \langle u, t, r\rangle, o_{u'} | \sigma^*(u) = u', \mathcal{K}\}$.

These probabilities should be computed appropriately according to the background knowledge $\mathcal{K}$ that we consider for the adversary. In the next subsections, we compute these probabilities for two adversaries with different background knowledge:

- Adversary (I) whose knowledge of users' mobility is their geographical distribution over the regions, i.e., $\mathcal{K} \equiv \hat{\pi}$.
- Adversary (II) who is a stronger adversary and knows the users' probability of transition between the regions, i.e., $\mathcal{K} \equiv \hat{p}$.

We construct $\hat{\pi}$ and $\hat{p}$ from the users' actual traces. The element $\hat{\pi}_u(r)$ of $\hat{\pi}$ is calculated as the fraction of time instants when user u is in region r. The element $\hat{p}_u(r_i, r_j)$ of $\hat{p}$ is calculated as the fraction of transitions of user u to r_j over all time instants when u is in region r_i.

We perform analytic probability calculations, where we also use the conditional independence of observed events, given the actual events. In effect, we decompose the desired probability into basic parts that can be computed from known functions. As these calculations are made by the adversary in performing the attack, the basic parts need to be computable from functions known to the adversary.

3.1 Adversary (I)

De-anonymization

$$
\begin{aligned}
&\Pr\left\{o_{u'} | \sigma(u) = u', \hat{\pi}\right\} = \\
&= \prod_t \Bigg(\sum_{r \in \mathcal{R}} \sum_{x \in \{0,1\}} \underbrace{\Pr\left\{o_{u'}(t) | \boldsymbol{X_u(t)} = x, \boldsymbol{A_u(t)} = \langle u, t, r \rangle, \sigma(u) = u', \hat{\pi}\right\}}_{\text{LPPM - Obfuscation mechanism}} \\
&\qquad \cdot \underbrace{\Pr\left\{\boldsymbol{X_u(t)} = x | \boldsymbol{A_u(t)} = \langle u, t, r \rangle, \hat{\pi}\right\}}_{\text{Application}} \\
&\qquad \cdot \underbrace{\Pr\left\{\boldsymbol{A_u(t)} = \langle u, t, r \rangle | \hat{\pi}\right\} \equiv \hat{\pi}_u^{\tau}(r), t \in \tau}_{\text{Background Knowledge of the Adversary}} \Bigg)
\end{aligned}
\tag{13}
$$

De-obfuscation

$$
\begin{aligned}
&\Pr\left\{\boldsymbol{A_u(t)} = \langle u, t, r \rangle, o_{u'}(t) | \sigma^*(u) = u', \hat{\pi}\right\} = \\
&= \Pr\left\{o_{u'}(t) | \boldsymbol{A_u(t)} = \langle u, t, r \rangle, \sigma^*(u) = u', \hat{\pi}\right\} \\
&\quad \cdot \Pr\left\{\boldsymbol{A_u(t)} = \langle u, t, r \rangle | \hat{\pi}\right\} \\
&= \Bigg(\sum_{x \in \{0,1\}} \underbrace{\Pr\left\{o_{u'}(t) | \boldsymbol{X_u(t)} = x, \boldsymbol{A_u(t)} = \langle u, t, r \rangle, \sigma^*(u) = u', \hat{\pi}\right\}}_{\text{LPPM - Obfuscation mechanism}} \\
&\qquad \cdot \underbrace{\Pr\left\{\boldsymbol{X_u(t)} = x | \boldsymbol{A_u(t)} = \langle u, t, r \rangle, \hat{\pi}\right\}}_{\text{Application}} \Bigg) \\
&\quad \cdot \underbrace{\Pr\left\{\boldsymbol{A_u(t)} = \langle u, t, r \rangle | \hat{\pi}\right\} \equiv \hat{\pi}_u^{\tau}(r), t \in \tau}_{\text{Background Knowledge of the Adversary}}
\end{aligned}
\tag{14}
$$

3.2 Adversary (II)

In this case, the calculations can be simplified if we use two helper functions α and β, as defined below. In effect, the problem that the attacker faces is equivalent to estimating the hidden state of a Hidden Markov Process. In the context of Hidden Markov Processes, the functions α and β are the forward-backward variables [16].

$$\alpha_t^{u,u'}(r) \equiv \Pr\left\{\boldsymbol{A_u(t)} = \langle u, t, r \rangle, o_{u'}(1), \cdots, o_{u'}(t) | \sigma(u) = u', \hat{p}\right\} \tag{15}$$

$$\beta_t^{u,u'}(r) \equiv \Pr\left\{o_{u'}(t+1), \cdots, o_{u'}(T) | \boldsymbol{A_u(t)} = \langle u, t, r \rangle, \sigma(u) = u', \hat{p}\right\} \tag{16}$$

In Appendix B, we show how to calculate these two functions in our case. Having calculated them for all $t \in \mathcal{T}$ and $r \in \mathcal{R}$, we can use them to compute the probabilities of interest.

De-anonymization

$$\Pr\left\{o_{u'} | \sigma(u) = u', \hat{p}\right\} = \sum_{r \in \mathcal{R}} \alpha_T^{u,u'}(r) \tag{17}$$

De-obfuscation

$$\Pr\left\{\boldsymbol{A_u(t)} = \langle u, t, r\rangle, o_{u'} | \sigma^*(u) = u', \hat{p}\right\} = \alpha_t^{u,u'}(r) \cdot \beta_t^{u,u'}(r) \tag{18}$$

where we compute α and β given σ^*.

4 Evaluation

In this section, we present the effectiveness of some location-privacy preserving mechanisms in protecting users' location privacy while they expose their location through some location-based applications. We evaluate the location privacy of users with respect to the two adversary types we introduced in the previous sections. We have extended the *Location-Privacy Meter* tool [19] by adding the location-based applications, implementing new LPPMs, and new localization attacks for sporadic applications, as described in the paper.

4.1 Simulation Setting

The location traces that we use in our simulation belong to $N = 20$ randomly chosen mobile users (vehicles) from the epfl/mobility dataset at CRAWDAD [15]. The area within which users move (the San Francisco bay area) is divided into $M = 40$ regions forming a 5×8 grid.

We evaluate various LPPMs that operate on top of two kinds of applications. The first type of application is the *once-in-a-while* application, which also serves as a baseline for comparison. In this type of application, events are exposed independently at random with the same probability θ. That is,

$$\Pr\left\{X_u(t) = 1 | A_u(t) = \langle u, t, r\rangle\right\} = \Pr\left\{X_u(t) = 1\right\} = \theta. \tag{19}$$

The second type of application is the *local search* application. In this application, users make queries, thus exposing their location, when they find themselves in unfamiliar places (which are the places that the user does not visit often, and hence needs more information about). We model this application as exposing the events of user u at location r independently at random with probability that is a decreasing function of $\pi_u(r)$. In particular,

$$\Pr\left\{X_u(t) = 1 | A_u(t) = \langle u, t, r\rangle, \pi\right\} = \theta(1 - \pi_u(r)). \tag{20}$$

where θ here determines the upper-bound on the probability of location exposure.

We refer to the application simply by using its parameter θ, and its type (o: once-in-a-while application, and s: local search). For example a local search application with exposure rate 0.1 is denoted by $\mathrm{APP}(0.1, \mathrm{s})$.

For our considered LPPMs, we have to define two modes of behavior, according to whether the application exposes or hides the location. When the application exposes the user's location, the LPPM obfuscates it by removing some low-order bits/digits of the location-stamp of the event. We refer to the number of removed bits as the *obfuscation level* ρ of the LPPM. When the application hides the location, the LPPM chooses, with some probability ϕ, to create a fake location and then obfuscates it (as it does for the actual locations). We consider two ways in which the LPPM can create a fake location: The first way is to create a fake location uniformly at random among all locations $r \in \mathcal{R}$, and the second way is to create it according to the aggregate user geographical distribution $\bar{\pi} = \frac{1}{N} \sum_{u \in \mathcal{U}} \pi_u$ (i.e., the average mobility profile). We refer to an LPPM using its parameters ϕ and ρ, and its type (u: uniform selection, g: selection according to the average mobility profile). For example $\mathrm{LPPM}(0.3, 2, \mathrm{u})$ injects a fake location (uniformly selected at random) with probability 0.3 if there is no location exposure, and obfuscated the (both fake and actual) locations by dropping their 2 low-order bits.

The metric that we use to evaluate the LPPMs is the expected error, as described in Section 2.5. We evaluate the effect of the application and LPPM parameters that we listed above (obfuscation level, probability ϕ of injecting a fake location) as well as the effect of the different application types and of the different ways of creating fake locations.

We are also interested in the effect of the pseudonym lifetime on the privacy of users. In our model, we consider that all users keep their pseudonyms from time 1 to T. By attacking at time T, we can compare the privacy achieved by users for various values of T.

4.2 Simulation Results

We run the simulator for all combinations of the following parameters: $\mathrm{APP}(0.1, \{\mathrm{o}, \mathrm{s}\})$, $\mathrm{LPPM}(\{0, 0.3, 0.6\}, \{0, 2, 4\}, \{\mathrm{u}, \mathrm{g}\})$, and pseudonym lifetimes $\{31, 71, 141, 281\}$. We then perform the de-anonymization and localization attacks (for both (I) weak, and (II) strong adversaries) that are described in the previous section. The results are averaged over 20 simulation runs. Hereafter, we present some of the results that we obtain regarding the anonymity and location-privacy of users.

In Figure 1, we plot user anonymity as a function of pseudonym lifetime. The anonymity is quantified as the percentage of users that are incorrectly de-anonymized by the attacker. Notice that we do not yet plot the location privacy of users, just their anonymity as defined. Each of the four sub-figures corresponds to each of the four combinations of adversary type (I-weak, II-strong) and LPPM type (u, g). Each line in a sub-figure corresponds to different combinations of obfuscation level and probability of injecting a fake location.

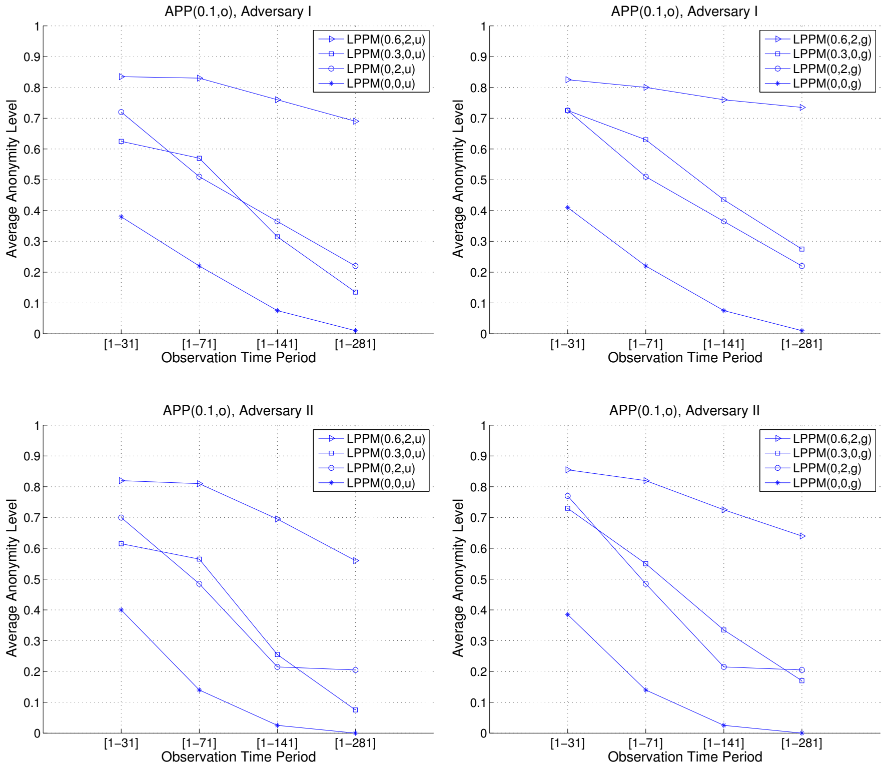

Fig. 1. User anonymity versus pseudonym lifetime in location-based application APP(0.1, o). The anonymity is quantified as the percentage of users that are incorrectly de-anonymized by the attacker. In the top two sub-figures, we consider the weak adversary (I), whereas in the bottom two, we consider the strong one (II). The left column considers the uniform (u) LPPM type, whereas the right column considers the LPPM type g. Each line in a sub-figure corresponds to different combinations of obfuscation levels {0, 2} and fake-location injection rates {0, 0.3, 0.6}.

We observe that the anonymity decreases as the pseudonym lifetime (the size of the observation period) increases. The same trend is seen in all four subfigures, for all combination parameters. By comparing the results that are obtained from different LPPMs, we observe the following interesting phenomenon, regarding the effect of stronger LPPM parameters, in particular when both the obfuscation level and the fake injection probability are non-zero: By jointly increasing the protection level of the two mechanisms, not only the absolute value of anonymity gets higher, but also the robustness to longer pseudonym lifetimes becomes better. That is, the level of anonymity drops with a slower rate as the pseudonym lifetime increases. This shows the relation between the effects of obfuscation and anonymization techniques. The LPPM designer can choose appropriately the parameters to achieve a desired level of anonymity; or alter-

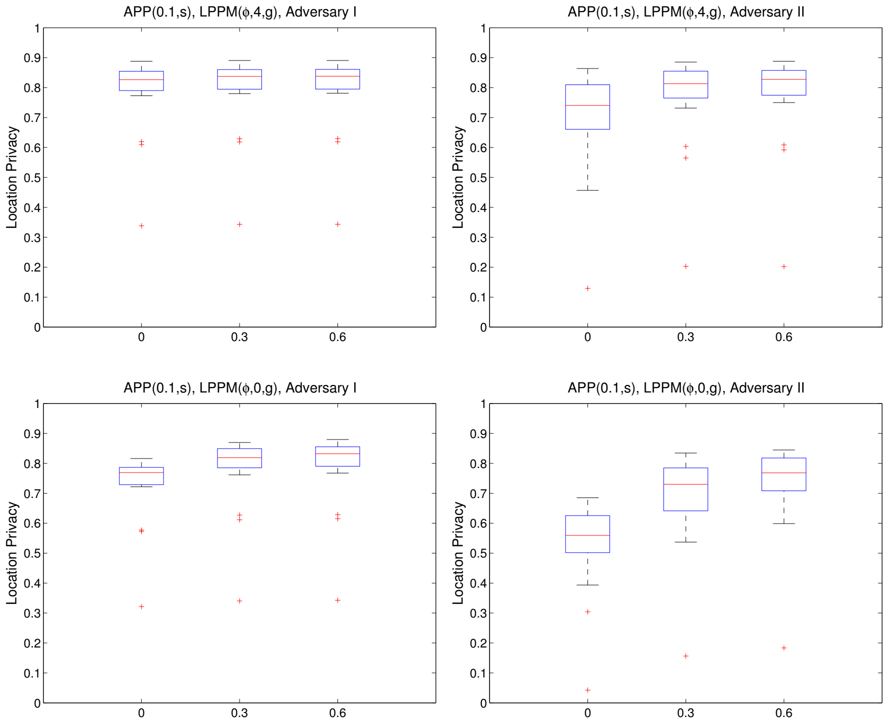

Fig. 2. Users' location privacy in location-based application $\mathrm{APP}(0.1, \mathrm{s})$, using various LPPMs, with respect to localization attack performed by two adversaries (I-weak: left column, and II-strong: right column). The x-axis shows the fake-location injection rate ϕ. The sub-figures corresponds to LPPM with obfuscation level 4 (for the top two), and 0 (for the bottom two). Each box-and-whisker diagram (boxplot) shows all location-privacy values (hence, system-level), where the bottom and top of a box show the 25th and 75th percentiles, and the central mark shows the median. The ends of the whiskers represent the most extreme data points not considered as outliers, and the outliers are plotted individually.

natively, the pseudonym should be changed when the desired level of anonymity is no longer achieved.

In Figure 2, we show the location privacy of users who (i) sporadically expose their location with exposure rate 0.1 in a *local search* application, and (ii) use LPPM that adds fake locations to their observed trace according to the *aggregate user geographical distribution*. As it is expected, the users' location privacy increases when the level of location-obfuscation or fake-location injection increases. However, the main finding of our result is that, in sporadic applications, the fake-location injection can dominate the obfuscation method, in preserving users' location-privacy, when the injection rate is higher. Moreover, adding fake location has a high impact on misleading the stronger adversary, as it reduces his success down to that of weaker adversary (compare the location-privacy improvement obtained by injecting fake-locations with rate 0.3 in the bottom sub-figures).

5 Related Work

The work related to our paper is threefold: (i) The papers that evaluate the risk of exposing locations through location-based services (which are mainly sporadic), (ii) The papers that aim at protecting users' location privacy for sporadic applications, and (iii) The papers that provide a framework for location privacy and describe possible threats and protections mechanisms as well as the location-privacy metrics.

The risk of location disclosure in mobile networks is evaluated in multiple papers. The authors use different attacks to de-anonymize the users' exposed traces (which are exposed in a sporadic manner). Ma *et al.* [14] make use of maximum likelihood estimation to identify the users from which the adversary has obtained some noisy past traces. Freudiger *et al.* [5] assume the adversary has access to the users' home and work addresses and performs the de-anonymization attack on the observed traces using some clustering algorithms. Similar de-anonymization of mobile users through identifying their home and work addresses have been performed by various researchers. Golle and Partridge [7], Beresford and Stajano [2], Hoh *et al.* [10], and Krumm [12] use different techniques to show that users can be identified by inferring where they spend most of their time (notably their home and workplace). De Mulder *et al.* [3] also present some statistical inference attacks on users' traces in GSM cellular networks. The authors show how easily the adversary can identify users if he has access to their location pattern (i.e., how they are distributed throughout the cells) in such setting. Compared to this set of contributions, in this paper we take two more major steps: We not only formalize the location-based application, but also the protection mechanisms that can be used to preserve users' location-privacy. Moreover, besides the de-anonymization, we evaluate the success of the adversary in finding the location of users over time. We provide a systematic formal framework that can be used to model the combination of a variety of LBSs and LPPMs.

Protecting location privacy of users in location-based services has received a tremendous attention from researchers in different disciplines such as database, and ubiquitous computing. A majority of the protection mechanisms revolve around combination of anonymization and location obfuscation. Duckham and Kulik [4] propose a formal model for location obfuscation techniques such as adding inaccuracy, imprecision, and vagueness. Krumm [12] shows that the effects of spatial cloaking algorithms and adding Gaussian noise, or discretizing the location (i.e., reducing granularity) can degrade the identification success of the adversary. Gruteser and Grunwald [8] propose spatial and temporal cloaking methods to increase the adversary's uncertainty in identifying the users. The privacy of users is quantified according to k-anonymity. Gedik *et al.* [6] propose an architecture and some algorithms to protect location privacy using personalized k-anonymity. A majority of the location-obfuscation techniques revolve around k-anonymity. The interested reader is referred to [20] for a more in depth overview of k-anonymity-based obfuscation techniques, and also to [19] for a quantitative analysis of k-anonymity metric for location privacy. As it is shown in [19, 20] these interesting approaches still lack an appropriate evaluation mechanism and metric that we provide in this paper. In addition to the obfuscation techniques, we also formalize and evaluate fake-location injection (adding dummy events) as another powerful method.

Krumm [13] provides a literature survey of computational location privacy. Shokri *et al.* [17] also provide a unified framework for location privacy, which is extended and more formalized in [19]. We have built up our system model on top of these frameworks by extending them in such a way that location-based services and new LPPMs can be defined and analyzed with respect to the localization attack.

6 Conclusion

We propose, to the best of our knowledge, the first formal framework for quantifying location privacy in the case where users expose their location sporadically. We formalize sporadic location-based applications. Using this formalization, we model various location-privacy preserving mechanisms, such as location obfuscation and fake-location injection. Formalizing both location-based applications and location-privacy preserving mechanisms in the same framework enables us to design more effective protection mechanisms that are appropriately tailored to each location-based service. We also establish an analytical framework, based on Bayesian inference in Hidden Markov Processes, to perform localization attacks on anonymized traces (for adversaries with different background knowledge). The results obtained from the simulations of the attacks on mobility traces unveil the potential of various mechanisms, such as the location obfuscation, the fake-location injection, and anonymization, in preserving location-privacy of mobile users.

Acknowledgements

The authors would like to thank Ehsan Kazemi for his valuable comments on the submitted manuscript, and also Vincent Bindschaedler for his effective contribution to the development of the Location-Privacy Meter.

References

1. Location-Privacy Meter tool. Available online through http://people.epfl.ch/reza.shokri, 2011.
2. A. R. Beresford and F. Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 2(1):46–55, 2003.
3. Y. De Mulder, G. Danezis, L. Batina, and B. Preneel. Identification via location-profiling in gsm networks. In *WPES '08: Proceedings of the 7th ACM workshop on Privacy in the electronic society*, pages 23–32, New York, NY, USA, 2008. ACM.
4. M. Duckham and L. Kulik. A formal model of obfuscation and negotiation for location privacy. In *Proceedings of Pervasive Computing, Third International Conference, PERVASIVE*, Munich, Germany, 2005.
5. J. Freudiger, R. Shokri, and J.-P. Hubaux. Evaluating the privacy risk of location-based services. In *Financial Cryptography and Data Security (FC)*, 2011.
6. B. Gedik and L. Liu. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *IEEE Transactions on Mobile Computing*, 7(1):1–18, 2008.
7. P. Golle and K. Partridge. On the anonymity of home/work location pairs. In *Pervasive '09: Proceedings of the 7th International Conference on Pervasive Computing*, pages 390–397, Berlin, Heidelberg, 2009. Springer-Verlag.
8. M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *MobiSys '03: Proceedings of the 1st international conference on Mobile systems, applications and services*, pages 31–42, New York, NY, USA, 2003. ACM.
9. B. Hoh and M. Gruteser. Protecting location privacy through path confusion. In *SECURECOMM '05: Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks*, pages 194–205, Washington, DC, USA, 2005. IEEE Computer Society.
10. B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. Enhancing security and privacy in traffic-monitoring systems. *IEEE Pervasive Computing*, 5(4):38–46, 2006.
11. B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. Preserving privacy in gps traces via uncertainty-aware path cloaking. In *CCS '07: Proceedings of the 14th ACM conference on Computer and communications security*, pages 161–171, New York, NY, USA, 2007. ACM.
12. J. Krumm. Inference attacks on location tracks. In *In Proceedings of the Fifth International Conference on Pervasive Computing (Pervasive), volume 4480 of LNCS*, pages 127–143. Springer-Verlag, 2007.
13. J. Krumm. A survey of computational location privacy. *Personal Ubiquitous Comput.*, 13(6):391–399, 2009.
14. C. Y. Ma, D. K. Yau, N. K. Yip, and N. S. Rao. Privacy vulnerability of published anonymous mobility traces. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, MobiCom '10, pages 185–196, New York, NY, USA, 2010. ACM.
15. M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser. CRAWDAD data set epfl/mobility (v. 2009-02-24). Downloaded from http://crawdad.cs.dartmouth.edu/epfl/mobility, Feb. 2009.
16. L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
17. R. Shokri, J. Freudiger, and J.-P. Hubaux. A unified framework for location privacy. Technical Report EPFL-REPORT-148708, EPFL, Switzerland, 2010.

18. R. Shokri, J. Freudiger, M. Jadliwala, and J.-P. Hubaux. A distortion-based metric for location privacy. In *WPES '09: Proceedings of the 8th ACM workshop on Privacy in the electronic society*, pages 21–30, New York, NY, USA, 2009. ACM.
19. R. Shokri, G. Theodorakopoulos, J.-Y. L. Boudec, and J.-P. Hubaux. Quantifying location privacy. In *IEEE Symposium on Security and Privacy*, Oakland, CA, USA, 2011.
20. R. Shokri, C. Troncoso, C. Diaz, J. Freudiger, and J.-P. Hubaux. Unraveling an old cloak: k-anonymity for location privacy. In *Proceedings of the 9th annual ACM workshop on Privacy in the electronic society*, WPES '10, pages 115–118, New York, NY, USA, 2010. ACM.
21. C. Troncoso, B. Gierlichs, B. Preneel, and I. Verbauwhede. Perfect matching disclosure attacks. In *Proceedings of the 8th international symposium on Privacy Enhancing Technologies*, PETS '08, pages 2–23, Berlin, Heidelberg, 2008. Springer-Verlag.

A Notations

Throughout the paper, we use bold capital letters to denote random variables, lower case letters to denote realizations of random variables, and script letters to denote sets within which the random variables take values. For example, a random variable $\boldsymbol{X}$ takes values x in $\mathcal{X}$.

$\mathcal{U}$	set of mobile users
$\mathcal{R}$	set of regions that partition the whole area
$\mathcal{T}$	time period under consideration
$\mathcal{A}$	set of all possible actual traces
$\mathcal{X}$	set of all possible exposed-locations bit-masks
$\mathcal{O}$	set of all observable traces
$\mathcal{U}'$	set of user pseudonyms
$\mathcal{R}'$	set of location pseudonyms (it is equivalent to $\mathcal{P}(R)$)
N	number of users
M	number of regions
T	number of considered time instants (length of $\mathcal{T}$)
a_u	actual trace of user u
x_u	exposed trace-bit-mask of user u
$o_{u'}$	observed trace of a user with pseudonym $u' \in \mathcal{U}'$
$\Delta(.,.)$	distortion (distance) function
p_u	actual mobility profile of user u
$\hat{p}_u$	profile of user u estimated by the adversary
π_u	geographical distribution of user u's location
$\hat{\pi}_u$	estimation of π_u by the adversary
$\mathcal{K}$	background knowledge of the adversary about users
$APP(\theta, \text{type})$	LBS application with location exposure rate θ, and types: o (once-in-a-while), and s (local search).
$LPPM(\phi, \rho, \text{type})$	LPPM with fake-location injection rate ϕ, obfuscation level ρ, and types: u (uniform selection of fake locations), and g (selecting the fake location from the aggregated geographical distribution of users).

Table 1. Notations

B Computing $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$

The computations of α (15) and β (16) are done recursively as follows.

$$
\begin{aligned}
&\alpha_1^{u,u'}(r) = \Pr\left\{\boldsymbol{A_u(1)} = \langle u,1,r\rangle, o_{u'}(1) | \sigma(u) = u', \hat{p}\right\} = \\
&= \sum_{x\in\{0,1\}} \underbrace{\Pr\left\{o_{u'}(1) | \boldsymbol{A_u(1)} = \langle u,1,r\rangle, \boldsymbol{X_u(1)} = x, \sigma(u) = u', \hat{p}\right\}}_{\text{LPPM - Obfuscation mechanism}} \\
&\qquad \cdot \underbrace{\Pr\left\{\boldsymbol{X_u(1)} = x | \boldsymbol{A_u(1)} = \langle u,1,r\rangle, \hat{p}\right\}}_{\text{Application}} \\
&\qquad \cdot \underbrace{\Pr\left\{\boldsymbol{A_u(1)} = \langle u,1,r\rangle | \hat{p}\right\} \equiv \hat{\pi}_u^{\tau}(r), t \in \tau}_{\text{Background Knowledge of the Adversary}}
\end{aligned}
\tag{21}
$$

$$
\begin{aligned}
&\alpha_{t+1}^{u,u'}(r) = \Pr\left\{\boldsymbol{A_u(t+1)} = \langle u,t+1,r\rangle, o_{u'}(1), \cdots, o_{u'}(t+1) | \sigma(u) = u', \hat{p}\right\} = \\
&= \sum_{x\in\{0,1\}} \underbrace{\Pr\left\{o_{u'}(t+1) | \boldsymbol{X_u(t+1)} = x, \boldsymbol{A_u(t+1)} = \langle u,t+1,r\rangle, \sigma(u) = u', \hat{p}\right\}}_{\text{LPPM - Obfuscation mechanism}} \\
&\qquad \cdot \underbrace{\Pr\left\{\boldsymbol{X_u(t+1)} = x | \boldsymbol{A_u(t+1)} = \langle u,t+1,r\rangle, \hat{p}\right\}}_{\text{Application}} \\
&\qquad \cdot \sum_{s\in\mathcal{R}} \underbrace{\Pr\left\{\boldsymbol{A_u(t+1)} = \langle u,t+1,r\rangle | \boldsymbol{A_u(t)} = \langle u,t,s\rangle, \hat{p}\right\} \equiv \hat{p}_u^{\tau_1,\tau_2}(s,r)}_{\text{Background Knowledge of the Adversary}} \\
&\qquad\quad \cdot \underbrace{\Pr\left\{\boldsymbol{A_u(t)} = \langle u,t,s\rangle, o_{u'}(1), \cdots, o_{u'}(t) | \sigma(u) = u', \hat{p}\right\}}_{\equiv \alpha_t^{u,u'}(s)}
\end{aligned}
\tag{22}
$$

$$
\beta_T^{u,u'}(r) = 1, \quad \forall r \in \mathcal{R} \tag{23}
$$

$$
\begin{aligned}
&\beta_t^{u,u'}(r) = \Pr\left\{o_{u'}(t+1), \cdots, o_{u'}(T) | \boldsymbol{A_u(t)} = \langle u,t,r\rangle, \sigma(u) = u', \hat{p}\right\} = \\
&= \sum_{s\in\mathcal{R}} \underbrace{\Pr\left\{o_{u'}(t+2), \cdots, o_{u'}(T) | \boldsymbol{A_u(t+1)} = \langle u,t+1,s\rangle, \sigma(u) = u', \hat{p}\right\}}_{\equiv \beta_{t+1}^{u,u'}(s)} \\
&\qquad \cdot \sum_{x\in\{0,1\}} \underbrace{\Pr\left\{o_{u'}(t+1) | \boldsymbol{X_u(t+1)} = x, \boldsymbol{A_u(t+1)} = \langle u,t+1,s\rangle, \sigma(u) = u', \hat{p}\right\}}_{\text{LPPM - Obfuscation mechanism}} \\
&\qquad\quad \cdot \underbrace{\Pr\left\{\boldsymbol{X_u(t+1)} = x | \boldsymbol{A_u(t+1)} = \langle u,t+1,s\rangle, \hat{p}\right\}}_{\text{Application}} \\
&\qquad\quad \cdot \underbrace{\Pr\left\{\boldsymbol{A_u(t+1)} = \langle u,t+1,s\rangle | \boldsymbol{A_u(t)} = \langle u,t,r\rangle, \hat{p}\right\} \equiv \hat{p}_u^{\tau_1,\tau_2}(r,s)}_{\text{Background Knowledge of the Adversary}}
\end{aligned}
\tag{24}
$$

where $t \in \tau_1$ and $t+1 \in \tau_2$.