A Heuristic Algorithm for Mobility-aware Location Obfuscation

Berker Ağır
EPFL, Switzerland
berker.agir@epfl.ch

Iris Safaka
EPFL, Switzerland
iris.safaka@epfl.ch

Malik Beytrison
EPFL, Switzerland
malik.beytrison@epfl.ch

Karl Aberer
EPFL, Switzerland
karl.aberer@epfl.ch

***Abstract*—Mobile users not only use on-demand location-based services increasingly (*e.g.,* checking in on online social networks), but also other mobile applications that provide a service based on location traces of users (*e.g.,* fitness tracking, health monitoring, etc.). This type of *continuous* tracking of user location introduces specific challenges to protection of location-privacy of mobile users. One of the challenges is ensuring the preservation of privacy levels of user location over time. Also, it is essential to build a location obfuscation area that results in high confusion for an adversary. In this paper, we address these challenges by proposing and evaluating a heuristic obfuscation algorithm that is mobility aware. Specifically, our heuristic algorithm reasons about a user's next location by taking into account user mobility history and direction of movement. Our experiments show that our approach outperforms a mobility-agnostic random obfuscation mechanism.**

I. Introduction

Users of mobile devices increasingly use services that are dominantly location-based. They not only share their location with their friends (*i.e.,* check-in on social networks), but also receive services based on continuous location tracking (*e.g.,* fitness and/or health monitoring). The downside of these services is that location information of users is contextually rich, and therefore can be used to infer private information such as political orientation, religious belief, etc., and in the worst case can be used for physical assault. The research community investigated the privacy risks of disclosing location information to untrusted parties extensively. For instance, Krumm [7], [8] showed how location traces of users can be inferred, even if they are obfuscated, and Golle *et al.* [5] demonstrated how home and work places of users can be pinpointed from them. Shokri *et al.* [13] also showed how an adversary can attack obfuscated user traces based on user mobility history.

Various approaches have been proposed to protect location privacy in location-based mobile systems by researchers [14]. A commonly adopted approach is to apply obfuscation on locations of users (*i.e.,* to deliberately degrade the quality of location information). However, even though location obfuscation introduces confusion for an adversary, studies also revealed some weaknesses of the approach in mobile applications where location data was continuously disclosed [1]. Against a reasoning adversary that has access to the geographical context and the mobility patterns of a user, simple obfuscation might prove to be an inadequate privacy-protection mechanism (PPM) and may result in reduced levels of location privacy in successive time steps, as the user moves.

In this paper, we tackle the challenge of protection location privacy in a continuous disclosure scenario. We propose a heuristic location-privacy protection mechanism that is aware of the user mobility when constructing the obfuscation area for user location in order to minimize the deterioration of location privacy over time. We call this heuristic approach *mobility-aware* location-privacy protection due to its awareness of user mobility history, direction of movement and speed of the user. We experimentally evaluate our heuristic mechanism and show that this approach provides a high level of location-privacy as compared to a random obfuscation mechanism against attackers both with and without knowledge on user history.

II. Framework

We consider mobile users equipped with smartphones moving in a geographical area that is discretized to M non-overlapping regions, *i.e.,* $\mathcal{R} = \{r_1, r_2, ..., r_M\}$ in discrete time space $\mathcal{T} = \{t_1, t_2, ..., t_N\}$. They send their location at every time instant $t_i \in \mathcal{T}$ to a server for, *e.g.,* receiving a location-based service or contributing to a sensing application. When sending her data, a user obfuscates her location. An obfuscated location $\mathcal{L}_t$ at time t is a set of locations from $\mathcal{R}$, ie $\mathcal{L}_t \subset \mathcal{R}$.

While performing location obfuscation, a user is revealing a set of c locations, *i.e.,* c regions, at each time instant t_i. We will be referring to the parameter c as the *location obfuscation parameter*. Clearly, the value of this parameter is determined by the level of privacy the user wishes for and there exists a trade-off between the utility of the application (location-based service or sensing application) and the user's level of privacy. Deciding on the location obfuscation parameter is out of the scope of this paper and we will be using a constant value for this parameter.

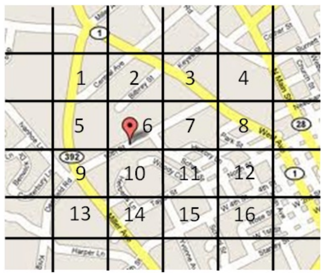
The following example summarizes the above. The area in Figure 1 is discretized into 16 regions and a user is moving in this area. At each time instant, she has to report c locations, out of which, one is her actual location and c-1 are fake. For example, on time instant t_i the user declares the set $\{5,6,7,9,10,11\}$ (Fig. 1b) instead of only 6 (Fig. 1a) which is her real position. In this example, the fake locations were

TABLE I: Table of Notations

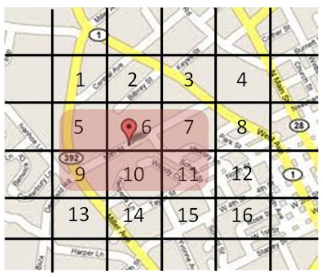
$c \geq 2$	location obfuscation parameter
$\mathcal{T} = \{0, 1, 2, \ldots, t\}$	set of time instants
$\mathcal{R} = \{r_1, r_2, \ldots, r_N\}$	set of N distinct regions in the area of interest
$\mathcal{L}_t \subset \mathcal{R}$	set of locations reported by the user at time t
$a(t) \in \mathcal{L}_t$	the actual location of the user at time t
$l_i \in \mathcal{L}_t$	the i_{th} fake location at time t, where $1 \leq i \leq c-1$
$\mathbf{neigh} : \mathcal{R} \rightarrow \mathcal{P}(\mathcal{R})$	function that gives the neighboring locations of a location
$\mathbf{Pr}_{r\rho} : \mathcal{R} \times \mathcal{R} \rightarrow [0, 1]$	the probability to go to region r from region ρ given by Equation 3

selected randomly. In Section IV, we will explain how to choose the fake locations so as to minimize the deterioration of privacy level at time instant t_{i+1}.

We use the notation in Table I in the rest of the paper.



(a) Actual location of a user



(b) Obfuscated location

Fig. 1: Location obfuscation example

A. Adversary Model

We assume an honest but curious adversary (potentially a service provider), *i.e.,* he will be able to observe the (obfuscated) locations of a user collected by a server and try to infer her actual trace from his observations, but never attempt to break protocols or hack otherwise to obtain more information. Furthermore, he may know a user's past traces (called background knowledge) and use them in his inference attacks to reduce his confusion. The intuition behind this is that people tend to have routines, hence regular mobility. The background knowledge the adversary has may or may not be complete. Additionally, he knows the maximum possible speed in terms of regions per time instant, at which a user can move. The adversary's goal is to infer the actual location of a user at each time instant by using his background information and the user's obfuscated trace by reasoning about her mobility.

B. User Mobility Model

In our setup, we are considering location obfuscation in successive time steps. Given the knowledge of the maximum speed, the past behavior and the direction of the user, the transitions between successive cells are characterized by probabilities. An adversary who has knowledge of these probabilities could reduce his uncertainty regarding the user's real location after observing her obfuscated location. The adversary can also benefit from road networks and maps of inaccessible locations when constructing such probabilities. For the sake of simplicity, we do not consider this type of knowledge, but our model is independent of it (*i.e.,* this type of knowledge can be easily integrated into the model). In this subsection we explain the user-mobility prediction models that we use in the design of the heuristic algorithm. We use three mobility models: a history-based, a direction-based and a combination of the two.

1) History-based Mobility Model: We adapt the human mobility model proposed by Calabrese *et al.* [2], which aims to predict a person's future location based on the individual's past behavior.

We denote the location of a user at time t as $a(t) = \rho$. The model predicts the user's next location $a(t+1)$ using past data. This is done by following a probabilistic approach: A probability is defined for each region $r \in \mathcal{R}$ to be the next location of the user as a function of the user's past behavior. We assume that the behavior of user is periodic over time with period T as modeled in [2]. More precisely, this probability is given by the formula:

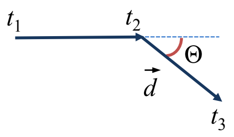
$$\mathbf{Pr}_h(a(t+1) = r | a(t) = \rho) = \frac{\sum_{m=1}^{\lfloor t/T \rfloor} f_h(a(t-Tm+1) = r | a(t-Tm) = \rho)}{\lfloor t/T \rfloor}, \forall r \in \mathcal{R} \tag{1}$$

where the frequency f_h on the right hand side is defined as:

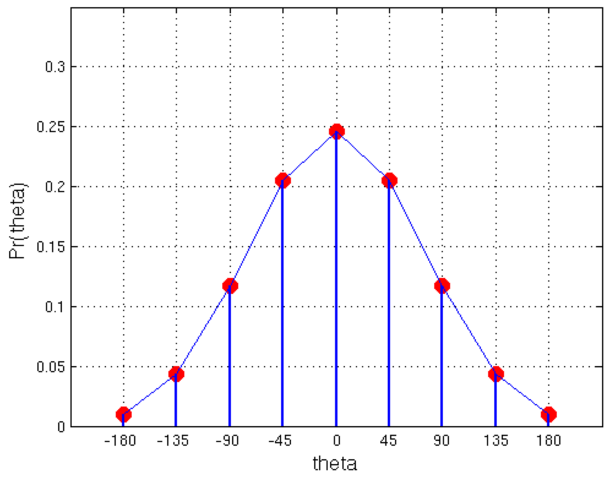
$$f_h(a(t+1) = r | a(t) = \rho) = \begin{cases} 1 & \text{if } a(t+1) = r \text{ and } a(t) = \rho \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

This model says that the probability of a region r to be the next destination of the user is equal to the frequency of visiting that region starting from region r_j during all previous periods $t - T + 1, t - 2T + 1, \ldots$. In [2], experimental results demonstrate a rather promising accuracy of this human-mobility model as compared to original traces used.

2) Direction-based Mobility Model: Direction-based mobility models are often used to model mobility in ad-hoc networks [3], since they are considered more realistic compared to fluid-flow or random-walk mobility models. The Gauss-Markov mobility model [9] falls into this category of models. Using this model, the mobile-user's next location is predicted based on the information gathered from the user's last location report, velocity and direction. We adopt a simple version of the model: The idea is that it is more probable for a user to continue straight ahead rather than abruptly turning back while



(a) User direction vector $\vec{d}$ with its angle Θ w.r.t. previous direction.



(b) The probability distribution over direction change angle Θ showing that retaining the direction has the highest probability. Please note that turning back has the lowest, yet a non-zero probability.

Fig. 2: The direction-based model that considers the angle of direction change in user movement.

moving. The following example aims to give an intuition of this model.

Consider a user at time t and let $\vec{d}$ be the vector of her velocity, *i.e.,* his direction, at time t as shown in Figure 2. The probability $\mathbf{Pr}(\Theta)$ that the user will change her direction by $\Theta \in [-180°, 180°]$ degrees at time $t+1$ takes values with respect to a normal distribution. Finally $\mathbf{Pr}_d(a(t+1) = r|a(t) = \rho, a(t-1) = \rho') = \mathbf{Pr}(\Theta)$, where the coordinates of regions $\rho \in \mathcal{R}$ and $\rho' \in \mathcal{R}$ are used to identify the direction $\vec{d}$ at time t, *i.e.,* we need the last two visited regions to determine the movement direction of the user.

3) The Combined Model: In this Section, we define a model to predict a user's behavior as a combination of the history-based model and the direction-based model:

$$\mathbf{Pr}_{r\rho}(a(t+1) = r|a(t) = \rho) = \alpha \cdot \mathbf{Pr}_h + (1-\alpha) \cdot \mathbf{Pr}_d \quad (3)$$

where $\alpha \in [0, 1]$ is the combination parameter and can change over time to model occasions when the user's behavior is more likely to be accurately predicted by her history data, if these are enough and available, and occasions where the direction-based model is better suited to predict future movement of the user.

III. Problem Statement

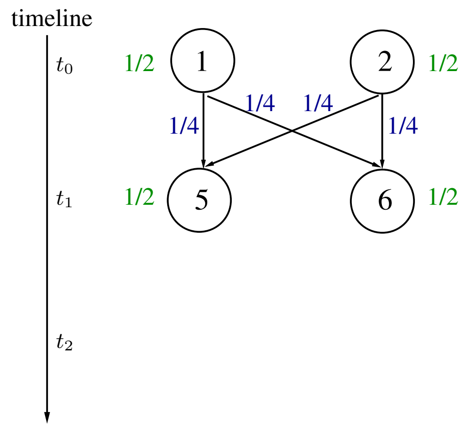
Before presenting the heuristic algorithm, we state the algorithm design problem through an example. For demonstration reasons we use a directed *linkability graph* as shown in Figure 3, where the vertices are labeled after the reported locations at each time instant and a link between two vertices exists if a transition is possible between them in successive time instants. Each vertex is assigned a *presence* probability $\mathbf{Pr}(a(t) = r)$, where r is also the label of the vertex representing region r. Each link is assigned a transition probability $\mathbf{Pr}(a(t) = r|a(t-1) = \rho) \cdot \mathbf{Pr}(a(t-1) = \rho) \neq 0$, where ρ is the origin vertex and r the destination (zero if the link does not exist). Using Bayesian inference we can calculate that a location r is the real one as follows:

$$\mathbf{Pr}(a(t) = r) = \sum_{\rho \in \mathcal{L}_{t-1}} \mathbf{Pr}(a(t) = r|a(t-1) = \rho) \cdot \mathbf{Pr}(a(t-1) = \rho) \quad (4)$$

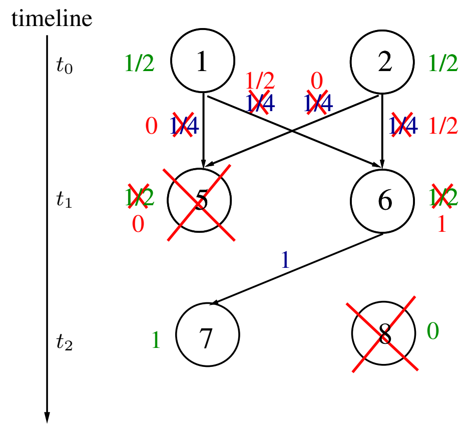
For the sake of simplicity, we assume in this section that transitions between vertices are equiprobable, therefore $\mathbf{Pr}(a(t) = r|a(t-1) = \rho)$ follows the uniform distribution, where $\sum_{r \in \mathcal{L}_t} \mathbf{Pr}(a(t) = r|a(t-1) = \rho) = 1, \forall \rho \in \mathcal{L}_{t-1}$. Also, the maximum speed of a user is one region per time instant. We will demonstrate that under these assumptions, a user, who obfuscates her location, can get decreased privacy levels in consecutive time instants. Consider the discretized area seen before and illustrated in Figure 1. We set the obfuscation parameter c to 2, the user moves within an area of 16 regions, her maximum speed is one cell per time unit and her trace is {2,6,7}. Ideally, for privacy protection, the linkability graph should not become disjoint and there should always an outgoing edge from every vertex in the previous time instants.

Let's assume that the user reports her actual location $a(t_0) = 2$ along with a fake one $l_1 = 1$ to the server at time t_0, thus $\mathbf{Pr}(a(t_0) = 2) = \mathbf{Pr}(a(t_0) = 1) = \frac{1}{2}$. At time t_1, the user reports $a(t_1) = 6$ and $l_1 = 5$ and we can compute the probability for each one of those being the real position using Equation 4, namely $\mathbf{Pr}(a(t_1) = 6) = \mathbf{Pr}(a(t_1) = 5) = \frac{1}{2}$. An adversary still has the highest uncertainty regarding the real position (Figure 3a). At t_2 the user chooses to report $a(t_2) = 7$ and $l_1 = 8$. Apparently, there are some links missing now, since region 7 is impossible to reach from region 5 and region 8 from both 5 and 6 in one time instant. An adversary has now information that would reduce his uncertainty: by exploiting the information at time t_2 and using Bayesian inference, he can recompute the probabilities at time instants t_0 and t_1. Applying Equation 4 to the new values she concludes that $\mathbf{Pr}(a(t_2) = 7) = 1$ and $\mathbf{Pr}(a(t_2) = 8) = 0$, meaning that he inferred the actual location of the user (see Figure 3b).

One can easily observe that the decrease of privacy level at time t_2 occurred due to the fact that the selection of region 8 as fake was done randomly. If the selection criterion was to select as fake a region that is a direct neighbor of both regions 5 and 6, then the linkability graph would not have become disjoint and no vertex would be removed, thus the



(a) High confusion for the adversary (*i.e.,* , high level of privacy)



(b) Low confusion for the adversary (*i.e.,* , low level of privacy)

Fig. 3: Example showing deterioration in privacy level

uncertainty would have remained high at time step t_2. This is a design specification that we take into account in the proposed heuristic algorithm of the next section.

Moreover, from the analysis above one can notice that deterioration of privacy level might occur due to transition probabilities assigned to links. A sophisticated adversary, that has knowledge of the geographical area and the mobility model of the user, can compute more accurate values of the conditional probability $\mathbf{Pr}(a(t) = r|a(t-1) = \rho)$ instead of simply assuming that they are equiprobable. For example, if a location r is not accessible by the user (even if ρ and r are direct neighbors), the probability $\mathbf{Pr}(a(t) = r|a(t-1) = \rho) = 0$, $\forall \rho \in \mathcal{L}_{t-1}$, resulting in a different value of $\mathbf{Pr}(a(t) = r)$. This is the second design specification taken into account in our heuristic algorithm.

IV. Mobility-aware Obfuscation Algorithm

In this section, we describe our heuristic obfuscation algorithm that is mobility-aware. The algorithm takes as input the user actual location $a(t)$, her current velocity v_t and the obfuscation parameter c, and returns the set $\mathcal{L}_t$ of c locations, representing the obfuscated location of the user. The algorithm checks the reachable locations from the last time instant w.r.t. v_t and by using the transition probabilities from the user mobility model, determines the $c-1$ fake locations to be included in $\mathcal{L}_t$ along with the actual location. The principle idea is that the algorithm chooses locations to populate $\mathcal{L}_t$ such that they will be the most probable locations along with $a(t)$ to go to from the locations in $\mathcal{L}_{t-1}$. Hence, the adversary's confusion will potentially be the maximum possible.

There are two base cases for our algorithm, namely for $t = 0$ and $t = 1$. For the sake of presentation and ease of understanding, we first explain the body of the algorithm for $t > 1$ and explain the base cases later. In summary, the algorithm consists of 3 steps. First, it determines the reachable locations at time t from the locations at time $t-1$ w.r.t. the velocity v_t. We call these locations the *candidate* locations $\mathcal{S}$ under consideration for $\mathcal{L}_t$. Secondly, the algorithm chooses $c-1$ locations from $\mathcal{S}$ such that they will have the highest transition probabilities w.r.t. $\mathcal{L}_{t-1}$. Finally the set $\mathcal{L}_t$ is formed by $a(t)$ and the determined $c-1$ locations and returned. The algorithm's pseudo-code is presented in Algorithm 1 with conditions $t = 0$ and $t = 1$ not included for readability purposes. Note that the algorithm has access to all past reported locations, *i.e.,* $\mathcal{L}_k$ for $k < t$, and user history.

Algorithm 1: Mobility-aware Obfuscation Algorithm

```
Input: $a(t)$, $c$, $v_t$
Output: $\mathcal{L}_t$ – the set of $c$ locations acting as the
        obfuscation area
1  $\mathcal{L}_t = \{a(t)\}$ ;
2  for each $\rho \in \mathcal{L}_{t-1}$ do
3      Find locations that can be reached from $\rho$,
       i.e., neigh$(\rho, v_t)$;
4  $\mathcal{I} = \bigcap_\rho$ neigh$(\rho, v_t)$ ;
5  $\mathcal{S} = (\mathcal{I} \setminus a(t)) \bigcap$ neigh$(a(t), v_t)$ ;
6  for each $r \in \mathcal{S}$ and each $\rho \in \mathcal{L}_{t-1}$ do
7      Compute the probability
       $\mathbf{Pr}_{r\rho}(a(t) = r|a(t-1) = \rho)$;
8  $i = 0$;
9  while $i < c-1$ do
10     $l_i = \arg\max_r \left(\sum_{\rho \in \mathcal{L}_{t-1}} \mathbf{Pr}_{r\rho}\right)$ ;
11     $\mathcal{L}_t = \mathcal{L}_t \bigcup \{l_i\}$ ;
12     $i = i + 1$ ;
13 return $\mathcal{L}_t$
```

At the beginning of the algorithm, $\mathcal{L}_t$ is initialized to $\{a(t)\}$ as it has to include the actual location of the users. Afterwards, in the first step (lines 2-5), the set $\mathcal{S}$ of candidate locations is determined by finding the set of neighbors of each location j in $\mathcal{L}_{t-1}$ with the limited range v_t, the velocity of the user. Then it finds the intersection of these sets to increase number of combinations of paths in the linkability graph to ensure maximum confusion for the adversary. The set $\mathcal{S}$ of candidate locations is finalized by first removing the actual location $a(t)$

from it and then intersecting it with the neighboring locations of $a(t)$. This last part filters the candidate locations to those within the proximity of the actual user location and also are accessible by all the locations in $\mathcal{L}_{t-1}$.

After determining the set $\mathcal{S}$, the transition probabilities from the locations reported previously, *i.e.,* in $\mathcal{L}_{t-1}$, to the ones in $\mathcal{S}$ are computed in the second step (lines 6-7). In the last step, $c - 1$ locations are chosen from $\mathcal{S}$ that provide the highest transition, hence presence, probabilities for time t and inserted to $\mathcal{L}_t$. Note that, in our experiments, we actually implement this step in a way that $\mathcal{L}_t$ consists of locations that form a joint polygon, in other words to avoid disjoint areas in $\mathcal{L}_t$. However, $\mathcal{L}_t$ may form areas of not regular shape (*i.e.,* square, rectangle, *etc.*).

We now explain the details for the base cases of our algorithm, *i.e.,* for $t = 0$ and $t = 1$. For $t = 0$, we do not have any transition to compute due to nonexistence of $\mathcal{L}_{t-1}$. Instead, the algorithm computes the presence probabilities of each location that is a neighbor of $a(0)$, replacing the lines 2-7. These probabilities can easily be computed from the history of user. Step 3 (lines 8-13) remains same except that instead of $\mathbf{Pr}_{r\rho}$ on line 10, the presence probabilities $\mathbf{Pr}(a(0) = k)$ are used. Secondly, for $t = 1$, we do have transitions, however, we cannot check a direction change because we need at least 3 points in space to be able to determine an angle of movement. Hence, we compute the transition probabilities $\mathbf{Pr}_{r\rho}$ based only on the history mobility model explained in Section II-B1 on line 7 of Algorithm 1.

Example: We now go briefly through the example presented in Section III again (where the trace of the user is {2, 6, 7} w.r.t. the area in Figure 1) in order to demonstrate the difference between the random selection of locations to report and our heuristic algorithm. We assume a maximum speed of 1 location per time instant with $c = 2$. At $t = 0$ with $a(0) = 2$, the set $\mathcal{S}$ is populated with all the one-hop distance neighbors of $a(0)$, *i.e.,* $\mathcal{S}$ = {1,2,3,5,6,7}. Here we have no access to past traces, so we just pick one location from the set $\mathcal{S}$ to report. Assume that we randomly choose to report location 1 as the fake location. Therefore $\mathcal{L}_0$ = {1,2}. For $t = 1$, we determine the candidate set $\mathcal{S}$ = {1,2,3,5,6,7} based on the previous locations reported. Again, we pick one out of these locations with uniform probability and report $\mathcal{L}_1$ = {5,6}. The algorithm's role becomes apparent now at $t = 2$. For every previously reported location, *i.e.,* in $\mathcal{L}_1$ = {5,6}, we find their neighbors, and intersect the two sets of neighbors to get the set $\mathcal{I}$. We have $\mathcal{I}$ = $\{1, 2, 5, 6, 9, 10\} \bigcap \{1, 2, 3, 5, 6, 7, 9, 10, 11\}$ = {1,2,5,6,9,10}. We then prepare $\mathcal{S}$ as described previously (*i.e.,* according to line 5 of the algorithm): $\mathcal{S}$ = {2,6,10}. Now that we have too many candidate locations to report, we have to compute the probability for all the elements of $\mathcal{S}$. But in this example we cannot compute the probability distribution so we just select a cell that could be reached from the previously reported locations, say cell 6. We finally report $\mathcal{L}_2$ = {6,7}. We can easily see that if we draw the linkability graph for this example we will get a fully connected graph (unlike with the random selection of locations), *i.e.,* we have no transition probability that is zero. Obviously, this is only a small example and not all the parameters are taken into account but it gives a good intuition of how the algorithm works and how it differs from a random obfuscation model.

V. Evaluation

We evaluate our heuristic algorithm by comparing it to a random obfuscation mechanism and using a dataset of real traces. We evaluate location privacy of users for both mechanisms using the Location-Privacy Meter developed by Shokri *et al.* [13], [12]. In the remainder of this section, we describe our dataset and methodology, explain the privacy metric we use and present our experimental results.

A. Dataset and Methodology

We use the real-world traces from the data collection campaign carried out by Nokia in Lausanne region [10] from 2009 to 2011. The area-of-interest we consider in Lausanne region is of size 1.25×1.0 km, which is discretized to 25×20 regions for computational limitations. We filter the dataset w.r.t. to this area and choose the users that have at least 15 chunks of 40-event long traces in this area. This results in a final set of 33 users. We train the adversary with the additional traces of each user while obfuscating and attacking one of them. We run our experiments for varying α and c values with heuristic and random obfuscation mechanisms separately. Finally, we attack all the generated obfuscated traces with two attackers (*i.e.,* using the Location-Privacy Meter), one with and one without the background knowledge on users' history.

B. Measuring Location Privacy

In our scenario, the adversary has access to the obfuscated trace of a user. His objective is to reconstruct the user's real trace based on this observation. The more accurately he succeeds in the reconstruction, the lower the level of location privacy that we obtain in our system. An effective metric for measuring location privacy is the *distortion-based* metric by Shokri *et al.* in [11]. This metric evaluates the expected error of the adversary in terms of expected distance between the inferred user location by the adversary and the actual user location.

In order to evaluate the effectiveness of our heuristic obfuscation algorithm, we used the Location-Privacy Meter (LPM) developed by Shokri *et al.* [13], [12]. This software is built based on a framework that formalizes the attack of the adversary, takes into account his background information on users' mobility patterns and calculates users' location-privacy protection levels based on the adversary's *accuracy*, *correctness* and *certainty* about users' actual trajectories. The LPM consists of several attack strategies based on Hidden Markov models. The observed traces serve as input to the tool as well as a distance function, and a transition matrix, that describes which transitions between regions are possible over consecutive time steps, given the geographical area and the

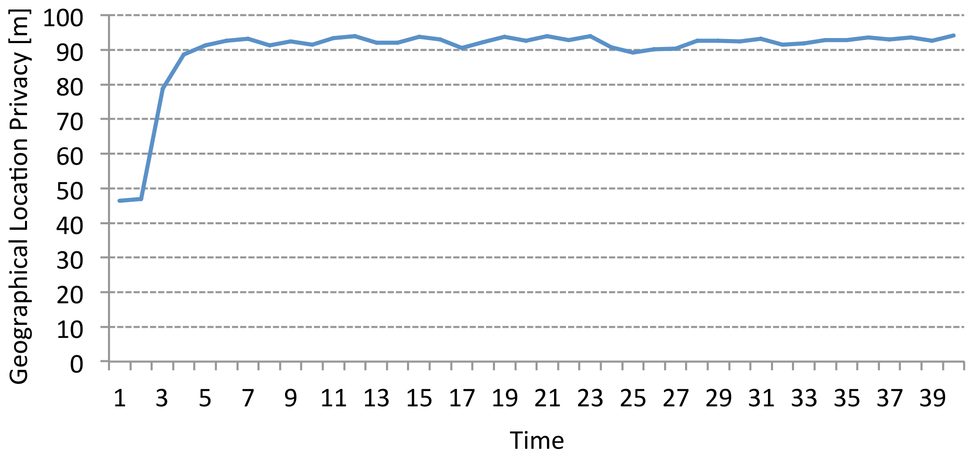


Fig. 4: Location privacy level in meters over time for one user, $c = 5$, $\alpha = 0.5$, against the weak adversary.

maximum speed of the user (in regions per time unit). The output is the level of location privacy in terms of expected distortion in meters at each time instant.

C. Experimental Results

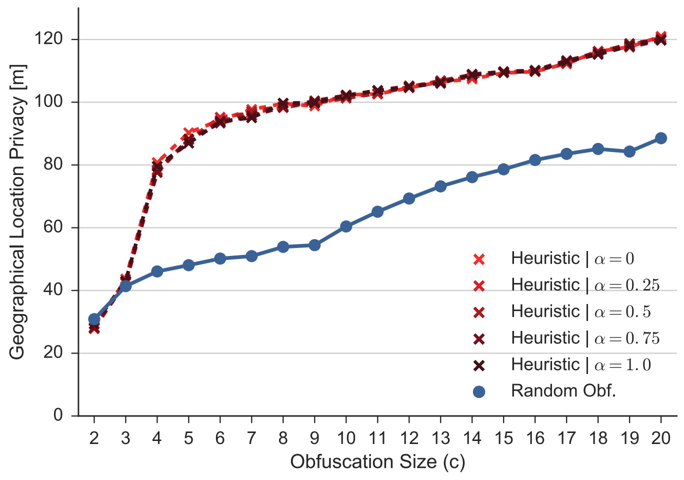


Fig. 5: Location privacy over the obfuscation parameter c averaged over all users and time instants, against the weak attacker.

In this section, we show various results obtained from our experiments. We ran our experiments with varying values of the obfuscation parameter c and the combination parameter α for our heuristic algorithm. We ran tests on the obfuscated traces with two different attackers (with and without background knowledge on the users' mobility). All location privacy levels are presented as the expected error of the adversary in meters a posteriori his inference. We refer to the attacker with background knowledge as the "strong attacker", and the one without background knowledge as the "weak attacker".

In Figure 4, we show the average location-privacy level of a user over time in meters for $c = 5$ and $\alpha = 0.5$ and against the weak attacker. We observe that the location privacy is relatively steady over time, except the first two time instants. This is due to the fact that we used uniform probabilities when choosing the fake locations for obfuscation in these time instants (*i.e.,* the base cases of the algorithm). But as of the third time instant, our actual heuristic algorithm shows its effect on the location privacy and we observe a jump around 100%. Also, as the algorithm can retain linkability among successive time instants, we see consistent privacy level protection over time. As discussed before, this is crucial for continuous location disclosure scenarios where users may accept a certain amount of loss in utility in order to meet a desired level of privacy.

We now show the impact of the parameters c and α on obfuscation. On Figure 5, we plot the average privacy levels over all users and all time instants over c obtained against the weak attacker. The results include privacy levels obtained with our heuristic algorithm for $\alpha = 0$, with $\alpha = 1$ and a random obfuscation algorithm that generates a randomly placed obfuscation area of size c. We observe that the value of α does not significantly change the results against the weak attacker. The results demonstrate the same trend for the strong attacker. However, we believe that the variance in results for different α values will increase for a richer dataset where user traces are longer and there are more diverse user mobility profiles. Additionally, we should be able to observe the impact of α clearer when users divert from their routines.

Finally, we compare the random obfuscation and our heuristic algorithm against the two attackers. Figure 6 shows the results of the algorithms for different values of obfuscation parameter c, and a comparison w.r.t. two adversary models (*i.e.,* the weak and the strong attackers). For the heuristic algorithm, we averaged the privacy levels with different values for α, (*i.e.,* for α = 0, 0.25, 0.5, 0.75 and 1) because the differences were negligible. The first observation on this plot is that the heuristic algorithm clearly outperforms the random one for $c \geq 4$. Even when the heuristic algorithm is attacked by the strong attacker, it still provides a better location-privacy

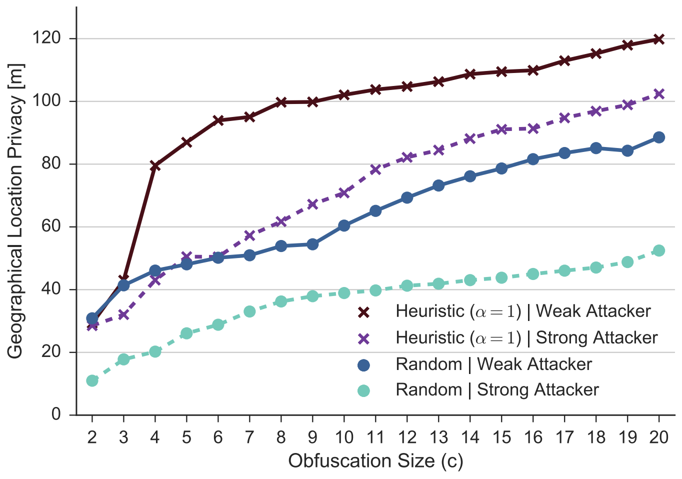


Fig. 6: Location privacy over the obfuscation parameter c averaged over all users and time instants for heuristic and random mechanisms against both attackers.

level than the random obfuscation mechanism against the weak attacker. This result demonstrates the importance of considering the mobility of the user when applying protecting location privacy. Obviously, the strong attacker obtains more information than the weak one for both heuristic and random mechanisms. However, the relative loss in the case of heuristic algorithm is considerably less than the loss in the case of random obfuscation. As expected, the location privacy is higher for larger values of c in all cases.

VI. Discussion

Note that our evaluation is under specific assumptions regarding adversary and even though we considered a strong adversary that knows rich user history, he is still computationally limited in our experiments. In this regard, the privacy levels we obtained in our results may be lower in reality if a more powerful adversary attacks the user traces. Nevertheless, we expect the privacy level difference observed between our heuristic algorithm and a random and static obfuscation algorithm to be still significant.

As a limitation, although trying to be realistic in the mobile setup has proved to provide good experimental results, this is not a formal proof that they constitute necessary and sufficient conditions. We believe working towards identifying these conditions and providing a formal proof is an interesting research direction. Furthermore, our heuristic algorithm can be extended to consider variable location-obfuscation size dynamically over time. This would provide an adaptive protection (hence privacy level), which might be desirable due to varying sensitivities of individuals.

Lastly, the obfuscation areas constructed by our algorithm may have irregular shapes and sometimes even consist of disjoint areas. This is not necessarily a shortcoming of our algorithm since it will always try to choose the regions that have the highest probabilities to confuse the adversary. On the other hand, there is room for improvement in terms of not visited locations that may have impact on the location privacy once visited in the future. We leave this dimension for future work.

VII. Related Work

The research community is aware of the privacy risks arising from mobility of users in continuous disclosure scenarios. There have been some attempts [15], [4] to address this issue by proposing velocity or mobility aware protection mechanisms. However, they fail to meet certain requirements which we addressed in this paper. Xu *et al.* [15] proposes to analytically consider a user's transition probability distributions among locations in order to derive certain obfuscation areas. They achieve this by constructing a linear program and solve it for the optimal solution for building a obfuscation area. Their idea is similar to our heuristic approach, yet they do not consider direction-based mobility in their system. Furthermore, they do not evaluate their protection mechanism against a powerful adversary with real traces. Instead, they attack either isolated user events or a short sequence of user events (*i.e.,* 2-3 events) and evaluate the privacy levels with the entropy metric.

Ghinita *et al.* [4] proposes a protection mechanism to protect location privacy in a velocity-aware way. Their model lacks user mobility history in protection mechanism. They also do not evaluate their mechanism's effectiveness against a localization attack with strong adversary assumptions. They do, however, consider sensitive semantic places on the map in order to determine the size and placement of a obfuscation area.

Götz *et al.* [6] propose a mobility-aware protection mechanism based on Hidden Markov models that take into account the knowledge of the adversary. Their proposed mechanism provide optimal solutions for keeping the confusion of the adversary high; however, the system requires an expensive initialization phase, thus requiring offloading some work to a remote server.

Finally, Agir *et al.* [1] proposed an adaptive protection mechanism for location privacy that considers an adversary's capabilities from user's side. They locally infer user location using Bayesian inference through whole user history in order to adapt the size of the obfuscation area used for protection. Our work differs from theirs in the obfuscation area construction: once they determine the size of an obfuscation area, they build a randomly generated rectangular area, whereas we build the obfuscation area using the mobility of user and the resulting obfuscation area is not necessarily a regular shape. In this sense, our work can be combined with theirs in order to enhance their adaptive protection approach and avoid increasing the obfuscation size too much, because our algorithm will be able to find a suitable obfuscation area for a given c much better.

VIII. Conclusion

In this paper, we explored how we can provide a powerful obfus-cation-based protection mechanism that is mobility-aware. We formulated the choice of locations used for building an obfuscation area at time instant t, such that the deterioration in the privacy level at time $t+1$ will be minimized while the privacy levels in the past are retained. We proposed a heuristic algorithm that takes into account the mobility of the user (*i.e.,* past behavior and the direction of movement) when applying obfuscation. The effectiveness of the heuristic algorithm was evaluated experimentally and the results are remarkable. Our motivation for a heuristic approach was to provide users with an efficient mechanism that can be adopted for resource-constrainted mobile devices. Moreover, our approach can run in near real-time and hence run in the background to protect users' location privacy unseemingly.

References

[1] B. Agir, T. G. Papaioannou, R. Narendula, K. Aberer, and J.-P. Hubaux. User-side Adaptive Protection of Location Privacy in Participatory Sensing. *Geoinformatica*, 18(1):165–191, 2014.

[2] F. Calabrese, G. Di Lorenzo, and C. Ratti. Human Mobility Prediction Based on Individual and Collective Geographical Preferences. In *Proc. of the IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2010.

[3] T. Camp, J. Boleng, and V. Davies. A Survey of Mobility Models for Ad hoc Network Research. *Wireless Communications and Mobile Computing*, 2(5):483–502, 2002.

[4] G. Ghinita, M. L. Damiani, C. Silvestri, and E. Bertino. Preventing Velocity-based Linkage Attacks in Location-aware Applications. In *Proc. of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2009.

[5] P. Golle and K. Partridge. On the Anonymity of Home/Work Location Pairs. In *Pervasive Computing*, volume 5538, pages 390–397. 2009.

[6] M. Götz, S. Nath, and J. Gehrke. Maskit: Privately Releasing User Context Streams for Personalized Mobile Applications. In *Proc. of the ACM SIGMOD International Conference on Management of Data*, 2012.

[7] J. Krumm. Inference Attacks on Location Tracks. In *Proc. of the International Conference on Pervasive Computing*, 2007.

[8] J. Krumm. A Survey of Computational Location Privacy. *Personal and Ubiquitous Computing*, 13(6):391–399, 2009.

[9] B. Liang and Z. Haas. Predictive Distance-based Mobility Management for PCS Networks. In *Proc. of the Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 1999.

[10] Nokia Research Center. Lausanne Data Collection Campaign. https://www.idiap.ch/dataset/mdc. Accessed: 2016-05-20.

[11] R. Shokri, J. Freudiger, M. Jadliwala, and J.-P. Hubaux. A Distortion-based Metric for Location Privacy. In *Proc. of ACM Workshop on Privacy in the Electronic Society (WPES)*, 2009.

[12] R. Shokri, G. Theodorakopoulos, G. Danezis, J.-P. Hubaux, and J.-Y. Le Boudec. Quantifying Location Privacy: The Case of Sporadic Location Exposure. In *Proc. of the Privacy Enhancing Technologies Symp. (PETS)*, 2011.

[13] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux. Quantifying Location Privacy. In *Proc. of IEEE Symposium on Security and Privacy (S&P)*, 2011.

[14] M. Wernke, P. Skvortsov, F. Dürr, and K. Rothermel. A Classification of Location Privacy Attacks and Approaches. *Personal and Ubiquitous Computing*, 18(1):163–175, 2014.

[15] J. Xu, X. Tang, H. Hu, and J. Du. Privacy-conscious Location-based Queries in Mobile Environments. *IEEE Transactions on Parallel and Distributed Systems*, 21(3):313–326, 2010.