Location Inference under Temporal Correlation

Yukun Dong*, Yidan Hu†, Aisha Aseeri‡, Depeng Li§, Rui Zhang*
*Department of Computer and Information Sciences, University of Delaware, Newark DE, 19716 USA
†Department of Computing Security, Rochester Institute of Technology, Rochester, NY 14623 USA
‡Department of Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia
§Department of Information and Computer Sciences, University of Hawaii at Manoa, Honolulu, HI 96822, USA
yukun@udel.edu, yidan.hu@rit.edu, aaaseeri@kau.edu.sa, depengli@hawaii.edu, ruizhang@udel.edu

***Abstract*—Location Based Services (LBSs) have become increasingly popular in the past decade, allowing mobile users to access location-dependent information and services. To protect user privacy while using LBSs, various Location Privacy Protection Mechanisms (LPPMs) have been proposed that obfuscate users' true locations through random perturbation. However, adversaries can still exploit the temporal correlation between a user's locations in multiple LBS queries to improve inference accuracy. In this paper, we introduce a novel location inference attack that strikes a good balance between inference accuracy and computational complexity by effectively exploiting temporal correlation. Simulation studies using synthetic and real datasets confirm the advantages of our proposed attack.**

***Index Terms*—Location privacy, inference attack, temporal correlation**

I. Introduction

In recent years, significant efforts have been made towards developing Location Privacy-Preserving Mechanisms (LPPMs) [1]–[4] that enable users to enjoy Location Based Services (LBSs) while safeguarding their location privacy. This is achieved by having users report an obfuscated location to the Location-Based Service Providers (LBSP). Previous studies on location privacy protection have typically considered two models: the sporadic model, where users access LBSs infrequently, and the continuous model, where users access LBSs periodically, and the real locations of the user in adjacent LBS queries commonly exhibit a certain degree of temporal correlation. This paper focuses on the continuous model, which includes LBSs such as continuous location sharing in social networks and periodically point-of-interest recommendation.

Although various LPPMs have been developed to address temporal correlation [5]–[9], the understanding of location inference attacks under temporal correlation is still limited. These attacks aim to uncover a user's actual trace from perturbed locations generated by an LPPM. While it is well-known that an attacker can leverage the temporal correlation among a user's locations to enhance inference accuracy, it is unclear how to fully exploit this correlation because the computational complexity of inferring a user's location trace from multiple queries grows exponentially with the length of location trace. Therefore, previous works [3], [4], [9], [10] assume that the adversary performs snapshot location inference attacks by independently estimating a user's true location. Efficiently utilizing temporal correlation to improve inference accuracy is still an open question.

This paper addresses the question of efficiently exploiting the temporal correlation among a user's adjacent LBS queries to improve location inference accuracy. We propose a novel generalized location inference attack based on three key ideas. Firstly, we leverage a recurrent relationship between inference results at two consecutive times to efficiently infer the user's location. Secondly, we only consider the user's recent perturbed locations within a fixed time window, reducing computational complexity. Thirdly, we limit our search to a small subset of candidate traces that are most likely to contain the user's true location trace. Our contributions include a significant reduction in computational complexity from exponential to polynomial with only a slight decrease in inference accuracy. Our contributions in this paper can be summarized as follows.

- To the best of our knowledge, we are the first to study efficient location inference under temporal correlations.
- We introduce a novel generalized inference attack for efficiently exploiting the temporal correlation among a user's adjacent LBS queries to strike a good balance between inference accuracy and computation cost.
- We conduct detailed simulation studies using synthetic and real location trace datasets to confirm the efficacy and efficiency of the proposed inference attack.

The rest of this paper is structured as follows. Section II introduces the system and adversary models. Section III presents the proposed generalized inference attack. Section IV reports the simulation results. Section V discusses the related work. We finally conclude this paper in Section VI.

II. Problem Formulation

In this section, we introduce the system and adversary models as well as our design goals.

A. System Model

We study an LBSP providing an LBS in a service area consisting of n discrete locations $\mathcal{X} = x_1, \ldots, x_n$. User submit LBS requests containing his/her current location to the LBSP periodically at each discrete time $t = 1, 2, \ldots$. We denote the user's real location at time t by $r^t \in \mathcal{X}$ and the real location trace from time i to j by $\boldsymbol{r}^{i,j} = (r^i, \ldots, r^j) \in \mathcal{X}^{j-i+1}$.

We model the user's mobility pattern as a memoryless Markov chain with an initial probability distribution π and a transition probability matrix M. The probability of the user's initial location being x_j is given by π_j for all $j = 1, \ldots, n$. The transition matrix M is an $n \times n$ matrix where each element $m_{i,j}$ is a conditional probability $p(r^{t+1} = x_j | r^t = x_i)$ that the user moves to location x_j at time $t+1$ given that they are at location x_i at time t. The probability of the user producing a trace $\boldsymbol{r}^t$ is given by $p(\boldsymbol{r}^t) = \pi(r^1) \cdot \prod_{i=2}^{t} p(r^i | r^{i-1})$.

To protect the location privacy, every user employs an LPPM to obfuscate the true location at each time t. We assume that the set of possible obfuscated locations is the same as $\mathcal{X}$. The LPPM f_t at time t maps a real location r^t to an obfuscated location o^t with probability $f_t(o^t | r^t)$ for all $r^t, o^t \in \mathcal{X}$. The expected loss in Quality of Service (QoS) caused by an LPPM is measured by $Q(f_t) = \sum_{r^t \in \mathcal{X}} \sum_{o^t \in \mathcal{X}} \pi(r^t) f_t(o^t | r^t) d(r^t, o^t)$, where $d(\cdot, \cdot)$ is a distance metric. We denote the obfuscated location trace from time i to j and from time 1 to j by $\boldsymbol{o}^{i,j} = (o^i, \ldots, o^j) \in \mathcal{X}^{j-i+1}$ and $\boldsymbol{o}^j = (o^1, \ldots, o^j) \in \mathcal{X}^j$, respectively. The probability of the user producing an obfuscated trace $\boldsymbol{o}^t$ given the real trace $\boldsymbol{r}^t$ is given by $p(\boldsymbol{o}^t | \boldsymbol{r}^t) = \prod_{i=1}^{t} f_i(o^i | r^i)$.

B. Adversary Model

We consider a passive adversary which may be either the LBSP itself or an external eavesdropper who can observe the obfuscated location in every LBS request from the target user. We assume that the adversary knows the initial probability distribution π, the transition probability matrix M, and the LPPM f_t employed at time t for all $t = 1, 2, \ldots$. The goal of the adversary is to infer user's location trace $\boldsymbol{r}^t$ from received $\boldsymbol{o}^t$. More specifically, the adversary carries out an inference attack, which can be viewed as a deterministic function $h(\cdot)$ that takes obfuscated location trace $\boldsymbol{o}^t$, LPPMs $f_1, \ldots, f_t$, the probability distribution of initial location π, and the transition probability matrix M as input and outputs an estimated location trace $\hat{\boldsymbol{r}}^t = (\hat{r}^1, \ldots, \hat{r}^t)$, i.e., $h(\boldsymbol{o}^t, f_1, \ldots, f_t, \pi, M) = \hat{\boldsymbol{r}}^t$.

C. Design Goals

We design the location inference attack to achieve the following two goals.

- *High inference accuracy*: The inference attack should infer user's true location trace with high accuracy.
- *High efficiency*: The inference attack should incur low computation cost.

We use the expected adversary error to measure the inference accuracy. Let $d(r^t, \hat{r}^t)$ be a function that measures the distance between the user's true location r^t and the adversary's estimate $\hat{r}^t = h(\boldsymbol{o}^t)$. The expected adversary error is defined as

$$\mathrm{E}(d(\boldsymbol{r}^t, \hat{\boldsymbol{r}}^t)) = \sum_{\boldsymbol{x}^t \in \mathcal{X}^t} \sum_{\boldsymbol{o}^t \in \mathcal{X}^t} p(\boldsymbol{x}^t) p(\boldsymbol{o}^t | \boldsymbol{x}^t) d(\boldsymbol{r}^t, \hat{\boldsymbol{r}}^t) \ , \quad (1)$$

where

$$d(\boldsymbol{r}^t, \hat{\boldsymbol{r}}^t) = \sum_{i=1}^{t} d(r^i, \hat{r}^i) \ . \quad (2)$$

III. A Generalized Inference Attack

In this section, we first present an optimal inference attack that can achieve the highest inference accuracy but incur high computational complexity as our baseline. We then give an overview of the proposed generalized inference attack and present its detailed operations.

A. An Optimal Inference Attack

In the optimal inference attack, the adversary estimates the user's location trace as a whole from all the obfuscated locations received so far at each time.

Specifically, at each time $t = 1, 2, \ldots$, the prior probability distribution of the user's location trace is given by

$$p(\boldsymbol{r}^t = \boldsymbol{y}^t) = \pi(y^1) \cdot \prod_{i=2}^{t} p(y^i | y^{i-1}) \ . \quad (3)$$

where $p(y^i | y^{i-1})$ is given by the transition matrix M.

Given LPPMs $f_1, \ldots, f_t$, the conditional probability of producing an obfuscated location trace $\boldsymbol{o}^t = (o^1, \ldots, o^t)$ from a true location trace $\boldsymbol{r}^t = (r^1, \ldots, r^t)$ is given by

$$p(\boldsymbol{o}^t | \boldsymbol{r}^t = \boldsymbol{y}^t) = \prod_{i=1}^{t} f_i(o^i | y^i) \ . \quad (4)$$

Given the obfuscated location trace $\boldsymbol{o}^t = (o^1, \ldots, o^t)$, the adversary computes the posterior distribution of the user's true location trace as

$$p(\boldsymbol{r}^t = \boldsymbol{y}^t | \boldsymbol{o}^t) = \frac{p(\boldsymbol{o}^t | \boldsymbol{r}^t = \boldsymbol{y}^t) p(\boldsymbol{r}^t = \boldsymbol{y}^t)}{\sum_{\boldsymbol{y}^t \in \mathcal{X}^t} p(\boldsymbol{o}^t | \boldsymbol{r}^t = \boldsymbol{y}^t) p(\boldsymbol{r}^t = \boldsymbol{y}^t)} \ . \quad (5)$$

Next, the adversary infers user's true location trace as the one with the minimal expected error. Specifically, if the adversary believes that the user's trace is $\boldsymbol{x}^t \in \mathcal{X}^t$, the expected error between the estimated trace and the real trace is given

$$E(\boldsymbol{x}^t | \boldsymbol{o}^t) = \sum_{\boldsymbol{y}^t \in \mathcal{X}^t} p(\boldsymbol{r}^t = \boldsymbol{y}^t | \boldsymbol{o}^t) d(\boldsymbol{y}^t, \boldsymbol{x}^t) \quad (6)$$

The adversary estimates the user's trace as the one with the minimal expected error, which is given by

$$\hat{\boldsymbol{r}}^t = \operatorname*{argmin}_{\boldsymbol{x}^t \in \mathcal{X}^t} E(\boldsymbol{x}^t | \boldsymbol{o}^t) \quad (7)$$

The optimal inference attack achieves the highest estimation accuracy as the adversary leverages all the available information to infer the user's true location trace. However, it also incurs the highest computational complexity. Specifically, the computational complexity comes from four steps of computation. First, we need to compute prior probability $p(\boldsymbol{r}^t = \boldsymbol{y}^t)$ for each $\boldsymbol{y}^t \in \mathcal{X}^t$ according to Eq. (3). There are total n^t terms, and computing each term takes $O(t)$ time, leading to a computational complexity of $O(tn^t)$. Second, we need to compute the conditional probability $p(\boldsymbol{o}^t | \boldsymbol{r}^t = \boldsymbol{y}^t)$ for each $\boldsymbol{y}^t \in \mathcal{X}^t$. There are total n^t terms, each takes $O(t)$ time. This also requires $O(tn^t)$. Third, we need to compute posterior probability $p(\boldsymbol{r}^t = \boldsymbol{y}^t | \boldsymbol{o}^t)$ for each $\boldsymbol{y}^t \in \mathcal{X}^t$, which requires $O(n^t)$. Finally, we need to compute $E(\boldsymbol{x}^t)$ according

to Eq. (6). There are n^t possible $\boldsymbol{y}^t$ and n^t possible $\boldsymbol{x}^t$. Both the multiplication and summation need $O(t)$. Thus, the overall complexity is $O(tn^{2t})$. In summary, the computational complexity of the optimal inference attack is $O(tn^{2t})$, which is exponential to time t and quickly becomes infeasible even for moderate n and t.

B. Overview

We design a generalized inference attack to greatly reduce the computational complexity of the optimal inference attack based on three key ideas.

First, we discover a recurrent relationship that can be exploited to reduce the computational complexity at each time t. Specifically, since

$$p(\boldsymbol{r}^t = \boldsymbol{y}^t | \boldsymbol{o}^t) = \frac{p(\boldsymbol{r}^t = \boldsymbol{y}^t, \boldsymbol{o}^t)}{p(\boldsymbol{o}^t)} \ ,$$

we can rewrite Eq. (7) as

$$\begin{aligned} \hat{\boldsymbol{r}}^t &= \operatorname*{argmin}_{\boldsymbol{x}^t \in \mathcal{X}^t} \sum_{\boldsymbol{y}^t \in \mathcal{X}^t} p(\boldsymbol{r}^t = \boldsymbol{y}^t | \boldsymbol{o}^t) d(\boldsymbol{y}^t, \boldsymbol{x}^t) \\ &= \operatorname*{argmin}_{\boldsymbol{x}^t \in \mathcal{X}^t} \sum_{\boldsymbol{y}^t \in \mathcal{X}^t} \frac{p(\boldsymbol{r}^t = \boldsymbol{y}^t, \boldsymbol{o}^t) \cdot d(\boldsymbol{y}^t, \boldsymbol{x}^t)}{p(\boldsymbol{o}^t)} \\ &= \operatorname*{argmin}_{\boldsymbol{x}^t \in \mathcal{X}^t} \sum_{\boldsymbol{y}^t \in \mathcal{X}^t} p(\boldsymbol{r}^t = \boldsymbol{y}^t, \boldsymbol{o}^t) \cdot d(\boldsymbol{y}^t, \boldsymbol{x}^t) \ , \end{aligned} \tag{8}$$

where the last equation holds because $p(\boldsymbol{o}^t)$ is the same for all $\boldsymbol{x}^t \in \mathcal{X}^t$. Let $E(\boldsymbol{x}^t) = \sum_{\boldsymbol{y}^t \in \mathcal{X}^t} p(\boldsymbol{r}^t = \boldsymbol{y}^t, \boldsymbol{o}^t) \cdot d(\boldsymbol{y}^t, \boldsymbol{x}^t)$ for all $\boldsymbol{x}^t \in \mathcal{X}^t$. We can further simplify Eq. (8) as

$$\hat{\boldsymbol{r}}^t = \operatorname*{argmin}_{\boldsymbol{x}^t \in \mathcal{X}^t} E(\boldsymbol{x}^t). \tag{9}$$

We now show how to compute $E(\boldsymbol{x}^t)$ efficiently by exploiting a recurrent relationship. Since $\boldsymbol{y}^t$ can be viewed as the concatenation of $\boldsymbol{y}^{t-1}$ and y^t, we can write it as $\boldsymbol{y}^t = \langle \boldsymbol{y}^{t-1}, y^t \rangle$. We then have

$$\begin{aligned} E(\boldsymbol{x}^t) &= \sum_{\boldsymbol{y}^t \in \mathcal{X}^t} p(\boldsymbol{r}^t = \boldsymbol{y}^t, \boldsymbol{o}^t) \cdot d(\boldsymbol{y}^t, \boldsymbol{x}^t) \\ &= \sum_{y^t \in \mathcal{X}} \sum_{\boldsymbol{y}^{t-1} \in \mathcal{X}^{t-1}} p(\boldsymbol{r}^{t-1} = \boldsymbol{y}^{t-1}, r^t = y^t, \boldsymbol{o}^t) \\ &\quad \cdot d(\langle \boldsymbol{y}^{t-1}, y^t \rangle, \boldsymbol{x}^t) \ . \end{aligned} \tag{10}$$

Let us define

$$\begin{aligned} E(\boldsymbol{x}^t, y^t) = &\sum_{\boldsymbol{y}^{t-1} \in \mathcal{X}^{t-1}} p(\boldsymbol{r}^{t-1} = \boldsymbol{y}^{t-1}, r^t = y^t, \boldsymbol{o}^t) \\ &\cdot d(\langle \boldsymbol{y}^{t-1}, y^t \rangle, \boldsymbol{x}^t) \end{aligned} \tag{11}$$

for all $\boldsymbol{x}^t \in \mathcal{X}^t$ and $y^t \in \mathcal{X}$. It follows that

$$E(\boldsymbol{x}^t) = \sum_{y^t \in \mathcal{X}} E(\boldsymbol{x}^t, y^t) \ . \tag{12}$$

Since $d(\boldsymbol{y}^t, \boldsymbol{x}^t) = d(\boldsymbol{y}^{t-1}, \boldsymbol{x}^{t-1}) + d(y^t, x^t)$, we can rewrite Eq. (11) as

$$\begin{aligned} &E(\boldsymbol{x}^t, y^t) \\ &= \sum_{\boldsymbol{y}^{t-1} \in \mathcal{X}^{t-1}} p(\boldsymbol{r}^{t-1} = \boldsymbol{y}^{t-1}, r^t = y^t, \boldsymbol{o}^t) d(\langle \boldsymbol{y}^{t-1}, y^t \rangle, \boldsymbol{x}^t) \\ &= \sum_{\boldsymbol{y}^{t-1} \in \mathcal{X}^{t-1}} p(\boldsymbol{r}^{t-1} = \boldsymbol{y}^{t-1}, r^t = y^t, \boldsymbol{o}^t) \cdot d(\boldsymbol{y}^{t-1}, \boldsymbol{x}^{t-1}) \\ &\quad + \sum_{\boldsymbol{y}^{t-1} \in \mathcal{X}^{t-1}} p(\boldsymbol{r}^{t-1} = \boldsymbol{y}^{t-1}, r^t = y^t, \boldsymbol{o}^t) \cdot d(y^t, x^t) \end{aligned} \tag{13}$$

Moreover, since

$$\begin{aligned} &p(\boldsymbol{r}^{t-1} = \boldsymbol{y}^{t-1}, r^t = y^t, \boldsymbol{o}^t) \\ &= p(\boldsymbol{r}^{t-1} = \boldsymbol{y}^{t-1}, \boldsymbol{o}^{t-1}) \cdot p(o^t, r^t = y^t | r^{t-1} = y^{t-1}) \\ &= p(\boldsymbol{r}^{t-1} = \boldsymbol{y}^{t-1}, \boldsymbol{o}^{t-1}) \cdot f_t(o^t | y^t) \cdot p(y^t | y^{t-1}) \ , \end{aligned} \tag{14}$$

we can rewrite the first term of Eq. (13) as

$$\begin{aligned} &\sum_{\boldsymbol{y}^{t-1} \in \mathcal{X}^{t-1}} p(\boldsymbol{r}^{t-1} = \boldsymbol{y}^{t-1}, r^t = y^t, \boldsymbol{o}^t) \cdot d(\boldsymbol{y}^{t-1}, \boldsymbol{x}^{t-1}) \\ &= \sum_{\boldsymbol{y}^{t-1} \in \mathcal{X}^{t-1}} p(\boldsymbol{r}^{t-1} = \boldsymbol{y}^{t-1}, \boldsymbol{o}^{t-1}) \cdot f_t(o^t | y^t) \cdot p(y^t | y^{t-1}) \\ &\quad \cdot d(\boldsymbol{y}^{t-1}, \boldsymbol{x}^{t-1}) \\ &= f_t(o^t | y^t) \sum_{y^{t-1} \in \mathcal{X}} p(y^t | y^{t-1}) \\ &\quad \cdot \sum_{\boldsymbol{y}^{t-2} \in \mathcal{X}^{t-2}} p(\boldsymbol{r}^{t-2} = \boldsymbol{y}^{t-2}, r^{t-1} = y^{t-1}, \boldsymbol{o}^{t-1}) \\ &\quad \cdot d(\boldsymbol{y}^{t-1}, \boldsymbol{x}^{t-1}) \\ &= f_t(o^t | y^t) \sum_{y^{t-1} \in \mathcal{X}} p(y^t | y^{t-1}) \cdot E(\boldsymbol{x}^{t-1}, y^{t-1}) \ . \end{aligned} \tag{15}$$

Let us further define

$$F(t-1, y^t) = \sum_{\boldsymbol{y}^{t-1} \in \mathcal{X}^{t-1}} p(\boldsymbol{r}^{t-1} = \boldsymbol{y}^{t-1}, r^t = y^t, \boldsymbol{o}^t) \tag{16}$$

for all $y^t \in \mathcal{X}$ and $t = 2, \ldots$ Therefore, the second term in Eq. (13) can be written as

$$\begin{aligned} &\sum_{\boldsymbol{y}^{t-1} \in \mathcal{X}^{t-1}} p(\boldsymbol{r}^{t-1} = \boldsymbol{y}^{t-1}, r^t = y^t, \boldsymbol{o}^t) \cdot d(y^t, x^t) \\ &\qquad = F(t-1, y^t) \cdot d(y^t, x^t) \ . \end{aligned} \tag{17}$$

Furthermore, since $p(\boldsymbol{r}^{t-1} = \boldsymbol{y}^{t-1}, r^t = y^t, \boldsymbol{o}^t) = p(\boldsymbol{r}^{t-1} = \boldsymbol{y}^{t-1}, \boldsymbol{o}^{t-1}) \cdot f_t(o^t | y^t) \cdot p(y^t | y^{t-1})$ according to Eq. (14), we

have

$$\begin{aligned} F(t-1,y^t) &= \sum_{\boldsymbol{y}^{t-1}\in\mathcal{X}^{t-1}} p(\boldsymbol{r}^{t-1}=\boldsymbol{y}^{t-1}, r^t=y^t, \boldsymbol{o}^t) \\ &= \sum_{\boldsymbol{y}^{t-1}\in\mathcal{X}^{t-1}} p(o^t,y^t|y^{t-1}) p(\boldsymbol{r}^{t-1}=\boldsymbol{y}^{t-1}, \boldsymbol{o}^{t-1}) \\ &= \sum_{y^{t-1}\in\mathcal{X}} f_t(o^t|y^t) p(y^t|y^{t-1}) \\ &\quad \cdot \sum_{\boldsymbol{y}^{t-2}\in\mathcal{X}^{t-2}} p(\boldsymbol{r}^{t-2}=\boldsymbol{y}^{t-2}, r^{t-1}=y^{t-1}, \boldsymbol{o}^{t-1}) \\ &= f_t(o^t|y^t) \sum_{y^{t-1}\in\mathcal{X}} p(y^t|y^{t-1}) \cdot F(t-2,y^{t-1}) \ . \end{aligned} \tag{18}$$

Finally, substituting Eq. (15) and (17) into Eq. (13), we get

$$\begin{aligned} E(\boldsymbol{x}^t,y^t) = f_t(o^t|y^t) &\sum_{y^{t-1}\in\mathcal{X}} p(y^t|y^{t-1}) E(\boldsymbol{x}^{t-1},y^{t-1}) \\ &+ d(y^t,x^t)\cdot F(t-1,y^t) \ . \end{aligned} \tag{19}$$

The recurrent relationship between $F(t-1,y^t)$ and $F(t-2,y^{t-1})$ shown in Eq. (18) and the recurrent relationship between $E(\boldsymbol{x}^t,y^t)$ and $E(\boldsymbol{x}^{t-1},y^{t-1})$ shown in Eq. (19) allow us to design a dynamic programming algorithm to compute $\{E(\boldsymbol{x}^t,y^t)|\boldsymbol{x}^t\in\mathcal{X}^t, y^t\in\mathcal{X}\}$ efficiently.

Second, we can further reduce the computational complexity of the optimal inference attack by limiting the number of past locations being considered. Intuitively, the temporal correlation between the user's two locations at different times decreases as more time has elapsed from the earlier location. It is thus reasonable to only consider up to a limited number of most recent obfuscated locations when inferring r^t.

Moreover, instead of examining every $\boldsymbol{x}^t\in\mathcal{X}^t$ in Eq. (8) to find the trace $\hat{\boldsymbol{r}}^t$ with the smallest expected error, we find that many traces with larger errors can be ruled out early. Therefore, we maintain a set of top k most likely candidate traces at every time for some positive constant k and only consider the candidate traces extended from the top k candidate traces obtained from the previous time.

Built upon the above three ideas, the general inference attack can achieve high inference accuracy with significantly lower computational complexity.

C. Detailed Procedures

We now introduce the detailed operations of the proposed inference attack at each time $t=1,2,\ldots$. Let $w\geq 1$ be a system parameter for the window size, i.e., only up to w most recent obfuscated locations will be used to infer user's current location.

1) At Time $t=1$.: On receiving the obfuscated location o^1 from the user, the adversary infers the user's true location as in the optimal inference attack.

First, for each $y^1\in\mathcal{X}$, the adversary computes

$$\begin{aligned} p(r^1=y^1,o^1) &= p(o^1|r^1=y^1)p(r^1=y^1) \\ &= f_1(o^1|y^1)\pi_1^-(y^1) \ , \end{aligned} \tag{20}$$

where $\pi_1^-=\pi$ denotes the initial prior distribution of user at each location.

Second, for each $x^1,y^1\in\mathcal{X}$, the adversary computes

$$\begin{aligned} F(1,y^1) &= p(r^1=y^1,o^1) \\ E(x^1,y^1) &= p(r^1=y^1,o^1)\cdot d(y^1,x^1) \end{aligned} \tag{21}$$

Third, for each $x^1\in\mathcal{X}$, the adversary computes the corresponding average adversary error as

$$E(x^1) = \sum_{y^1\in\mathcal{X}} E(x^1,y^1) \tag{22}$$

and estimates the user's location as

$$\hat{r}^1 = \operatorname*{argmin}_{x^1\in\mathcal{X}} E(x^1) \ . \tag{23}$$

In addition, the adversary computes the posterior distribution of user's location after observing o^1 as

$$\begin{aligned} p(r^1=y^1|o^1) &= \frac{p(o^1|r^1=y^1)p(r^1=y^1)}{\sum_{y^1\in\mathcal{X}} p(o^1|r^1=y^1)p(r^1=y^1)} \\ &= \frac{f_1(o^1|y^1)\pi_1^-(y^1)}{\sum_{y^1\in\mathcal{X}} f_1(o^1|y^1)\pi_1^-(y^1)} \ , \end{aligned} \tag{24}$$

for all $y_1\in\mathcal{X}$. Denote such posterior distribution as π_1^+, the updated prior probability can be calculated as

$$\pi_2^- = \pi_1^+ M \ . \tag{25}$$

Finally, the adversary finds the set of $k'=\min(k,n)$ locations in $\mathcal{X}$ that have the smallest average adversary error, denoted by $\mathcal{P}^{1,1}$, and records $\{(y^1,F(1,y^1))|y^1\in\mathcal{X}\}$, and $\{\langle x^1,y^1,E(x^1,y^1)\rangle|x^1\in\mathcal{P}^{1,1},y^1\in\mathcal{X}\}$ to facilitate inference at subsequent times, where k is a system parameter.

2) At Time $t=2,\ldots,w-1$.: At each time $t=2,\ldots,w-1$, the adversary carries out t concurrent and overlapping inference attacks. For each $j\in[1,t]$, the jth inference attack extends the inference attack from time j to $t-1$ to infer $\boldsymbol{r}^{j,t}$ based on $\mathcal{P}^{j,t-1}$ and newly observed o^t, and initiates the tth inference attack to infer r^t based on π_t^- and $\boldsymbol{o}^t$.

Consider the jth inference attack as an example, where $1\leq j\leq t$. The adversary first constructs a candidate trace set $\mathcal{C}^{j,t}$ from trace set $\mathcal{P}^{j,t-1}$ stored at time $t-1$ as

$$\mathcal{C}^{j,t} = \{(x^j,\ldots,x^{t-1},x^t)|(x^j,\ldots,x^{t-1})\in\mathcal{P}^{j,t-1}, x^t\in\mathcal{X}\} \ . \tag{26}$$

Since $|\mathcal{P}^{j,t-1}|=k'=\min(k,n^{t-j})$ and $|\mathcal{X}|=n$, we have $|\mathcal{C}^{j,t}|\leq kn$.

Second, for each $\boldsymbol{x}^{j,t}\in\mathcal{C}^{j,t}$, the adversary computes $E(\boldsymbol{x}^{j,t})$ from $\{(y^{t-1},F(t-j-1,y^{t-1}))|y^{t-1}\in\mathcal{X}\}$ and $\{\langle\boldsymbol{x}^{j,t-1},y^{t-1},E(\boldsymbol{x}^{j,t-1},y^{t-1})\rangle|\boldsymbol{x}^{j,t-1}\in\mathcal{P}^{j,t-1},y^{t-1}\in\mathcal{X}\}$ based on the recurrent relationship. Specifically, for each $y^t\in\mathcal{X}$, it computes $F(t-j,y^t)$ according to Eq. (18) as

$$F(t-j,y^t) = f_t(o^t|y^t) \sum_{y^{t-1}\in\mathcal{X}} p(y^t|y^{t-1})\cdot F(t-j-1,y^{t-1}) \tag{27}$$

For every pair of $\boldsymbol{x}^{j,t} \in \mathcal{C}^{j,t}$ and $y^t \in \mathcal{X}$, it further computes $E(\boldsymbol{x}^{j,t}, y^t)$ according to Eq. (19).

$$E(\boldsymbol{x}^{j,t}, y^t) = f_t(o^t|y^t) \sum_{y^{t-1} \in \mathcal{X}} p(y^t|y^{t-1}) \cdot E(\boldsymbol{x}^{j,t-1}, y^{t-1}) + d(y^t, x^t) F(t-j, y^t) \tag{28}$$

Moreover, for every $\boldsymbol{x}^{j,t} \in \mathcal{C}^{j,t}$, the adversary computes

$$E(\boldsymbol{x}^{j,t}) = \sum_{y^t \in \mathcal{X}} E(\boldsymbol{x}^{j,t}, y^t) \ . \tag{29}$$

Third, the adversary finds the set of $k' = \min(k, |\mathcal{C}^{j,t}|)$ location traces in $\mathcal{C}^{j,t}$ with the smallest average adversary error and stores them as $\mathcal{P}^{j,t}$. The adversary also stores $\{(y^t, F(t-j, y^t))|y^t \in \mathcal{X}\}$ and $\{\langle \boldsymbol{x}^{j,t}, y^t, E(\boldsymbol{x}^{j,t}, y^t)\rangle | \boldsymbol{x}^{j,t} \in \mathcal{P}^{j,t}, y^t \in \mathcal{X}\}$ to facilitate the inference attack at time $t+1$.

After performing the t inference attacks, the adversary obtains t sets of most likely traces $\mathcal{P}^{1,t}, \ldots, \mathcal{P}^{t,t}$. The adversary uses $\mathcal{P}^{1,t}$ to estimate the user's trace as

$$\hat{\boldsymbol{r}}^t = \underset{\boldsymbol{r}^t \in \mathcal{P}^{1,t}}{\operatorname{argmin}} E(\boldsymbol{x}^{1,t}) \ . \tag{30}$$

Let $\hat{\boldsymbol{r}}^t = \{\hat{r}^1, \ldots, \hat{r}^t\}$. The user's location at time t is estimated as $\hat{r}^t$. Note that $\mathcal{P}^{2,t}, \ldots, \mathcal{P}^{t,t}$ will be used to infer locations $\hat{r}^{w+1}, \ldots, \hat{r}^{w+t-1}$ at times $t^{w+1}, \ldots, t^{w+t-1}$, respectively.

In addition, the adversary can update the prior distribution by computing the posterior probability as

$$\begin{aligned} p(r^t = y^t|o^t) &= \frac{p(o^t|r^t = y^t)p(r^t = y^t)}{\sum_{y^t \in \mathcal{X}} p(o^t|r^t = y^t)p(r^t = y^t)} \\ &= \frac{f_t(o^t|r^t = y^t)\pi_t^-(y^t)}{\sum_{y^t \in \mathcal{X}} f_t(o^t|r^t = y^t)\pi_t^-(y^t)} \ , \end{aligned} \tag{31}$$

Denote such posterior distribution as π_t^+, the updated prior probability can be calculated as

$$\pi_{t+1}^- = \pi_t^+ M \ . \tag{32}$$

3) At Time $t \geq w$.: At each time $t \geq w$, the adversary carries out w concurrent and overlapping inference attacks in a similar way to time $1 < t < w$. For each $j \in [1, w]$, the jth inference attack extends the inference attack from time $t-w+j$ to $t-1$ to infer $\boldsymbol{r}^{t-w+j,t}$ based on $\mathcal{P}^{t-w+j,t-1}$ and newly observed obfuscated location o^t and initiates the jth inference attack to infer r^t based on π_t^- and o^t.

Consider the jth inference attack as an example, where $1 \leq j \leq w$. The adversary first constructs a candidate trace set $\mathcal{C}^{t-w+j,t}$ from trace set $\mathcal{P}^{t-w+j,t-1}$ stored at time $t-1$ as

$$\mathcal{C}^{t-w+j,t} = \{(x^{t-w+j}, \ldots, x^{t-1}, x^t) | (x^{t-w+j}, \ldots, x^{t-1}) \in \mathcal{P}^{t-w+j,t-1}, x^t \in \mathcal{X}\} \ . \tag{33}$$

Since $|\mathcal{P}^{t-w+j,t-1}| = k' = \min(k, n^{w-j})$ and $|\mathcal{X}| = n$, we have $|\mathcal{C}^{t-w+j,t}| \leq kn$.

Second, for each candidate trace $\boldsymbol{x}^{t-w+j,t} \in \mathcal{C}^{t-w+j,t}$, the adversary computes $E(\boldsymbol{x}^{t-w+j,t})$ from $\{(y^{t-1}, F(w-j-1, y^{t-1}))|y^{t-1} \in \mathcal{X}\}$ and $\{\langle \boldsymbol{x}^{t-w+j,t-1}, y^{t-1}, E(\boldsymbol{x}^{t-w+j,t-1}, y^{t-1})\rangle | \boldsymbol{x}^{t-w+j,t-1} \in \mathcal{P}^{t-w+j,t-1}, y^{t-1} \in \mathcal{X}\}$ based on the recurrent relationship. Specifically, for each $y^t \in \mathcal{X}$, it computes $F(w-j, y^t)$ according to Eq. (18) as

$$F(w-j, y^t) = f_t(o^t|y^t) \sum_{y^{t-1} \in \mathcal{X}} p(y^t|y^{t-1}) \cdot F(w-j-1, y^{t-1}) \tag{34}$$

For every pair of $\boldsymbol{x}^{j,t} \in \mathcal{C}^{t-w+j,t}$ and $y^t \in \mathcal{X}$, it further computes $E(\boldsymbol{x}^{t-w+j,t}, y^t)$ according to Eq. (19).

$$E(\boldsymbol{x}^{t-w+j,t}, y^t) = f_t(o^t|y^t) \sum_{y^{t-1} \in \mathcal{X}} p(y^t|y^{t-1}) \cdot E(\boldsymbol{x}^{t-w+j,t-1}, y^{t-1}) + d(y^t, x^t) F(w-j, y^t) \tag{35}$$

Moreover, for every $\boldsymbol{x}^{t-w+j,t} \in \mathcal{C}^{t-w+j,t}$, the adversary computes

$$E(\boldsymbol{x}^{t-w+j,t}) = \sum_{y^t \in \mathcal{X}} E(\boldsymbol{x}^{t-w+j,t}, y^t) \ . \tag{36}$$

Third, for each $j \in [1, w]$, the adversary finds the set of $k' = \min(k, |\mathcal{C}^{t-w+j,t}|)$ location traces in $\mathcal{C}^{t-w+j,t}$ with the smallest average adversary errors and store them as $\mathcal{P}^{t-w+j,t}$.

After performing the w inference attacks, the adversary obtains w sets of most likely traces $\mathcal{P}^{t-w+1,t}, \ldots, \mathcal{P}^{t,t}$. The adversary uses $\mathcal{P}^{t-w+1,t}$ to estimate the user's trace as

$$\hat{\boldsymbol{r}}^{t-w+1,t} = \underset{\boldsymbol{r}^{t-w+1} \in \mathcal{P}^{t-w+1,t}}{\operatorname{argmin}} E(\boldsymbol{x}^{t-w+1,t}) \ . \tag{37}$$

Let $\hat{\boldsymbol{r}}^{t-w+1} = \{\hat{r}^{t-w+1}, \ldots, \hat{r}^t\}$. The user's location at time t is estimated as $\hat{r}^t$. Note that $\mathcal{P}^{t-w+2,t}, \ldots, \mathcal{P}^{t,t}$ will be used to infer locations $\hat{r}^{t+1}, \ldots, \hat{r}^{t+w-1}$ at times $t+1, \ldots, t+w-1$, respectively.

Finally, the adversary updates the prior distribution by computing the posterior probability according to Eq. (31) and computes the updated prior probability according to Eq. (32).

D. computational complexity

We now analyze the computational complexity of the proposed generalized inference attack. 1) At time $t = 1$, there are n possible x^1 and y^1, thus computing $E(x^1)$ requires $O(n^2)$ complexity. 2) At each time $t \geq 2$, there are at most w concurrent and overlapping attack. For each attack j, we need to first construct candidate trace set $\mathcal{C}$, which requires $O(n)$ time. Second, we compute $F(w-j, y^t)$, which requires $O(n^2)$ complexity as there are n possible y^t and y^{t-1}. Then, we need to compute $E(\boldsymbol{x}^{t-w+j}, y^t)$. Since there are at most kn possible $\boldsymbol{x}^t$ and n possible y^t, each combination takes $O(n)$ time, this process requires $O(kn^3)$ time. Finally, we compute $E(\boldsymbol{x}^{t-w+j,t})$, which requires $O(kn^2)$ time. In summary, the computational complexity for each inference attack is $O(kn^3)$. Since there are at most w inference attack, the complexity at each time is $O(kwn^3)$. Compared with the optimal attack, the computational complexity of generalized inference attack is greatly reduced.

TABLE I: Default Settings

Parameter	Value	Description
n	100	# of locations
	10	The length of the user's location trace
w	3	The size of sliding window
k	10	
$\bar{H}$	0.2	The normalized entropy rate of M
α	0.4	The privacy parameter of *LH*
ϵ	1	The privacy parameter of *Exp*
n_r	10	# of real traces
n_o	40	# of obfuscated traces

IV. Performance Evaluation

In this section, we evaluate the performance of the proposed location attacks via extensive simulation studies using both synthetic and real datasets. All simulations are done in MATLAB on a PC with 2.90 GHz Intel i5 CPU and 8 GB memory.

A. Datasets

1) Synthetic Dataset: We generate several synthetic datasets with $n = 100$ locations and uniform prior distributions but different Markov transition matrices that represent different temporal correlations among the user's locations at different times. Specifically, since the Markov transition matrix characterizes the mobility pattern of users, i.e., how a user moves from one location to another between different times, the temporal correlation of the mobility pattern can be measured by the entropy rate [11] of the Markov chain defined by the transition matrix M, which is a well known metric. In particular, the entropy rate of a Markov chain is defined by

$$H(M) = -\sum_i \mu_i \sum_j m_{ij} \log(m_{ij}) \,, \tag{38}$$

where m_{ij} is the probability of a user moves from location x_i to x_j, μ_i is the stationary distribution satisfying $\mu_i = \sum_j \mu_j m_{ji}$, and $\sum_{i=1}^{n} \mu_i = 1$. We further adopt a normalization process to enforce the entropy rate in the range of $[0, 1]$. The normalized entropy rate is calculated as

$$\bar{H}(M) = H(M)/\log_2(n) \,. \tag{39}$$

The smaller the normalized entropy rate of the transition matrix, the larger the temporal correlation among the user's different locations, and vice versa. For example, if $m_{i,j} = \frac{1}{n}$ for all $1 \le i, j \le n$, then the normalized entropy rate of M is 1, which indicates that there is no temporal correlation between any two locations of a user. As another example, if for every row $i \in [1, n]$, there exist $i_j \in [1, n]$ such that $m_{i,i_j} = 1$ and $m_{i,j} = 0$ for all $j \neq i_j$, then the normalized entropy rate of M is 0, which indicates the location at the next time is completely determined by the current one. For a given entropy rate, we generate the transition matrix M using the algorithm in [12].

2) Real Dataset: Gowalla [13] is a location-based social networking website where users share their locations by checking-in. We take the data covering most of San Francisco region with latitude (37.5500, 37.8010) and longitude (-122.5153, -122.3789). For simplicity, we further split the area into 15×10 cells and choose the centers of these cells as the set of locations. We count the number of transitions from x_i to x_j for any two locations in the dataset and normalize it to compute the m_{ij} for the Markov transition matrix. The normalized entropy rate of the transition matrix for this dataset is 0.1365. For the prior distribution π, we count the number of user's check-ins at each cell and normalize the resulting histogram.

B. Simulation Settings

We evaluate the performance of the proposed location inference attacks on two LPPM instantiations.

- *Local Hashing* (LH) [9]: For a user at location x_i, the user reports the true location $r = x_i$ with probability α or one of the eight neighboring locations of x_i with probability $(1 - \alpha)/8$.
- *Exponential LPPM* (Exp) [14]: For a user at location x_i, he reports an obfuscated location o with probability proportional to $\exp(-d(x_i, o) \cdot \epsilon)$, where ϵ the privacy budget indicating the level of privacy protection. We also consider the same domain of original locations and obfuscated locations and 8-connected neighboring locations as in *LH*.

Given a dataset with prior distribution π and transition matrix M, we first randomly generate $n_r = 10$ real traces. For each real trace $\boldsymbol{r}^t$, we randomly generate $n_o = 40$ obfuscated traces using *LH* or *Exp*. Given an obfuscated trace $\boldsymbol{o}^t$, the attack infers the real trace using a specific inference mechanism, $h(\cdot)$, i.e., $\hat{\boldsymbol{r}}^t = h(\boldsymbol{o}^t)$. Table I summarizes the default setting unless mentioned otherwise.

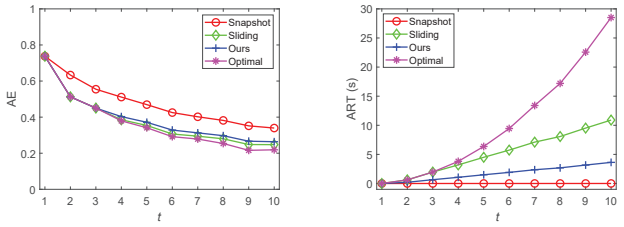
We compare the performance of the proposed inference attack with three other inference attacks.

- *Optimal inference attack* (Optimal): As mentioned in Sec. III-A, the optimal inference attack considers all received obfuscated locations to infer the user's true location trace as a whole. The inference accuracy achieved by Optimal can be viewed as the upper bound of any inference attacks.
- *Sliding window attack* (Sliding): It considers w most recent obfuscated locations for inference without limiting the search space at each time, which is a special case of the proposed inference attack with $k \to \infty$.
- *Snapshot attack* (Snapshot): It is another special case of the proposed inference attack with window size $w = 1$.

We use two metrics to evaluate the performance of attacks: *average adversary error (AE)* and *average running time (ART)*. We define AE as the average distance between a real trace and the estimated trace, which is given by

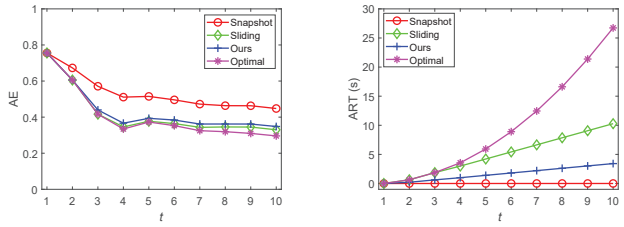
$$AE = \frac{1}{t n_r n_o} \cdot \sum_{\boldsymbol{r}^t} \sum_{h(\boldsymbol{o}^t)} d(\boldsymbol{r}^t, h(\boldsymbol{o}^t)) \,. \tag{40}$$

Moreover, we measure the computation cost of an inference attack by ART, which is defined as the average execution time needed to infer the user's location trace from a reported obfuscated trace.



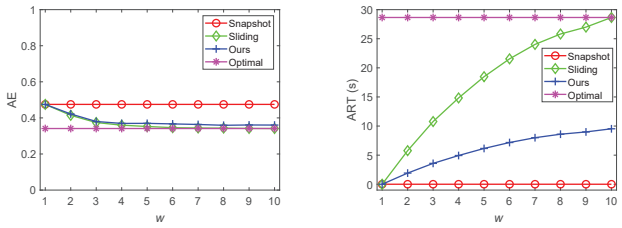
(a) AE vs. t (b) ART vs. t

Fig. 1: AE and ART vs. t under LH.



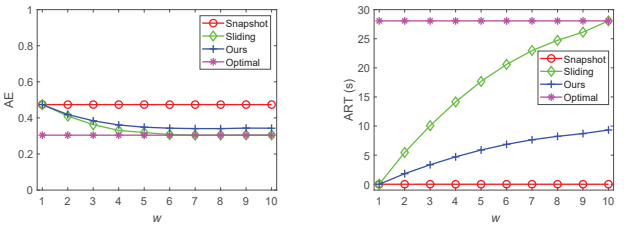
(a) AE vs. t (b) ART vs. t

Fig. 2: AE and ART vs. t under Exp.



(a) AE vs. w (b) ART vs. w

Fig. 3: AE and ART vs. w under LH.



(a) AE vs. w (b) ART vs. w

Fig. 4: AE and ART vs. w under Exp.

C. Simulation Results on Synthetic Datasets

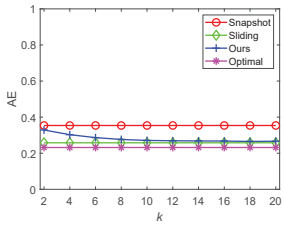
We first report the results on synethic datasets.

1) Impact of t: Fig. 1(a) compares four location inference attacks, namely Optimal, Sliding, Snapshot, and Ours, against LH as the trace length t increases from 1 to 10. All attack mechanisms exhibit a decrease in the adversary error (AE) as the trace length increases because longer traces provide more information for the adversary to infer the user's moving behavior accurately. Among the four attacks, Snapshot has the highest AE since it disregards the temporal correlation between the locations in a trace and infers each location independently based on the corresponding obfuscated location. In contrast, Optimal estimates the entire user's location trace from all the obfuscated locations received so far at each time, resulting in the smallest AE. Our proposed Ours outperforms Snapshot by utilizing the temporal correlation among the most recent w obfuscated locations to estimate each location in the trace, resulting in a much smaller AE close to the one under Optimal, especially for short traces. The inference accuracy of Ours is slightly lower than that of Optimal because considering only the most recent w obfuscated locations to infer each location may not capture the temporal correlation between locations beyond the w time slots. Additionally, the AE of Ours is slightly higher than that of Sliding since Ours only considers the k location traces with the smallest AEs during the most recent w time slots.

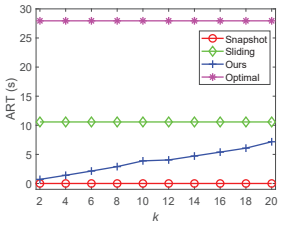
Fig. 1(b) illustrates the impact of trace length on computation cost, measured by ARTs, for four location inference attacks against LH. The ART of Snapshot is independent of trace length, remaining at approximately 0.003s. However, the ARTs of the other three attacks increase with trace length, as expected, since longer traces require more time to infer all locations. The ART of Optimal grows exponentially due to the exponentially increasing search space, whereas the ARTs of Sliding and Ours increase linearly with t. However, Ours has a significantly lower ART than Sliding since it has a much smaller search space.

Overall, compared with Optimal and Sliding, Ours can significantly reduce computation cost while sacrificing a slight amount of inference accuracy. In comparison with Snapshot, Ours can greatly improve inference accuracy, resulting in a much smaller AE, with a slight increase in computation cost. These results demonstrate that Ours can achieve a better balance between inference accuracy and computation cost than other attack mechanisms. Fig. 2 shows the AEs and ARTs under Snapshot, Optimal, Sliding, and Ours against the Exp mechanism, confirming the cost-effectiveness of the proposed attack against various LPPMs.

2) Impact of w: Fig.3 compares the AEs and ARTs of different attack mechanisms against LH mechanism with sliding window size w varying from 1 to 10. The AEs and ARTs of Snapshot and Optimal are not affected by w and are plotted for reference only. From Fig.3(a), we observe that the AEs of Sliding and Ours first decrease sharply and then gradually decrease until approaching that of Optimal. Initially, using more reported locations for each location estimation allows the adversary to leverage the temporal correlations among more locations for improved inference accuracy, i.e., a smaller AE. As w increases further, the additional past locations used for inference have limited effect on improving the inference accuracy, resulting in a slightly decreased or stable AE. Although Ours has a slightly higher AE than Sliding due to the limited search space, it outperforms Sliding in terms of ART by a large margin, especially when w is large, as shown in Fig. 3(b).

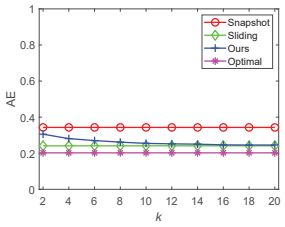


(a) AE vs. k

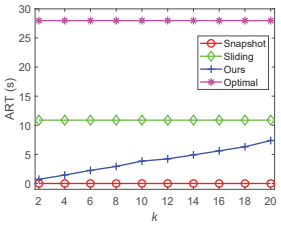


(b) ART vs. k

Fig. 5: AE and ART vs. k under LH.



(a) AE vs. k



(b) ART vs. k

Fig. 6: AE and ART vs. k under Exp.

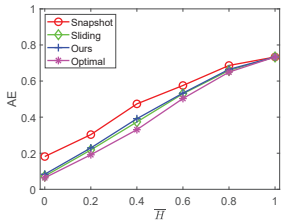
These results indicate that a relatively small sliding window size is good enough to have a high inference accuracy while maintaining a small computation cost. For example, when $w = 3$, the AE of Ours is about 0.02 higher than the one under Optimal, but the computation cost of Ours is only about 4s, which is much smaller than 28s introduced by Optimal and quite affordable in practice. Fig. 4 shows the impact of the sliding window size under the four attack mechanisms against the Exp mechanism. The performances of the four attack mechanisms are almost the same as the ones in Fig. 3, which confirm that Ours can not only achieve a better trade-off between inference accuracy and computation cost but also be practical in reality.

3) Impact of k: Fig. 5 and Fig. 6 compare the AEs and ARTs under Snapshot, Optimal, Sliding, and Ours with k increasing from 2 to 20, when the LPPM is LH and Exp, respectively. The AE and ART of Snapshot, Optimal, and Sliding are not affected by k and plotted as reference only. As we can see from Fig. 5(a) and Fig. 6(a), the AE of Ours initially decreases fast as k increases from 2 to 10 and then becomes stable or decreases slightly as the k increases from 10 to 20. The reason for the initial decrease is that when k is larger, the more candidate traces with the smallest AEs are considered, the more likely that the candidates contain user's true location, resulting in a smaller AE. As k further increases, it is very likely that the current candidate traces already include the user's true trace, which results in a lower chance to further decrease the AE by considering more candidate traces and thus have a slightly decreased or stable AE. From Fig. 5(b) and Fig. 6(b), we can see that the ART of Ours increases linearly as k increases, which is expected as the computational complexity $O(kwn^3)$ is linear to k. Moreover, considering Figs. 5(a) and 5(b) together, we can see that when k is large enough, e.g., $k = 8$, the AE of Ours is 0.276, which is almost the same with 0.258 under Sliding, but Ours can reduce the ART of Sliding by about 70%. These results demonstrate that Ours can dramatically reduce the computation cost while maintaining high inference accuracy by limiting the search within a small subset of candidate traces.

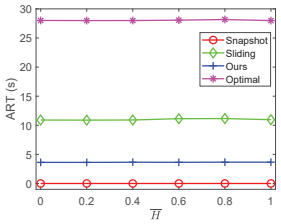
4) Impact of $\bar{H}$: Fig. 7(a) compares the AEs of four attack mechanisms with the normalized entropy rate varying from 0 to 1. As we can see, the AE increases as $\bar{H}$ increases for all attack mechanisms, which is anticipated as the larger the $\bar{H}$ of the Markov transition matrix, the weaker the temporal correlation between two adjacent locations, the more difficult to predict the next location based on the previous locations he observes, resulting in an increased AE. Moreover, Ours outperforms Snapshot with smaller AE, especially when $\bar{H}$ is very small, e.g., $\bar{H} = 0.2$, which indicates that Ours can take advantage of the temporal correlations among the user's locations to improve the inference accuracy. In particular, the four attack mechanisms have the same AEs when $\bar{H} = 1$, which is also anticipated. The reason is that when $\overline{H} = 1$, the user moves totally at random, and there is no temporal correlation between the users two locations. Thus considering a large window size or even all observed locations cannot improve the inference accuracy resulting in the same AE. From Fig. 7(b), we can see that the ARTs of the four attack mechanisms are not affected by $\bar{H}$, which is also anticipated as the user's behavior does not affect the inference procedures. We can also see that Ours has a very low ART, which is very close to the ideal one achieved by Snapshot but outperforms Optimal and Sliding with a large margin. From Fig. 8, we can see that the four attack mechanisms have similar performance with the one in Fig. 7, which indicates that Ours can achieve a better trade-off between inference accuracy and computation cost against different LPPMs.

These results indicate that the user's moving behavior has a great impact on the adversary's inference accuracy. The smaller the normalized entropy rate, the stronger the temporal correlations among the locations in a trace, the higher inference accuracy can be achieved by the advanced attack considering temporal correlation, and vice versa.

5) Impact of α and ϵ: We now study the performance of the attacks under different LPPMs. Fig. 9(a) compares the AEs and ARTs of the four attack mechanisms against LHs with α varying from 0 to 1. As we can see, the AEs of all attack mechanisms decrease as α increases. The reason is that under a LH with large α, the user would report the real location with a higher probability, i.e., a lower level of privacy protection, and thus the adversary can infer the true location with higher accuracy. In particular, when $\alpha = 1$, the user would directly report his/her true location without any privacy protection, and the adversary could infer the true location completely accurately, i.e., $AE = 0$. In addition, even when α is small, e.g., $\alpha = 0.2$, the AE of Ours is still small. From Fig. 10(a), we can see that the AEs of four attack mechanisms decrease as ϵ increases, which is expected as the higher the ϵ, the lower

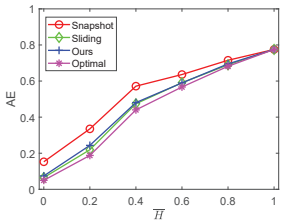


(a) AE vs. $\overline{H}$

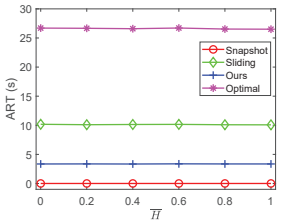


(b) ART vs. $\bar{H}$

Fig. 7: AE and ART vs. $\bar{H}$ under LH.

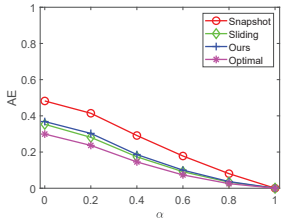


(a) AE vs. $\bar{H}$

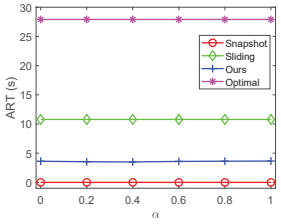


(b) ART vs. $\bar{H}$

Fig. 8: AE and ART vs. $\bar{H}$ under Exp.

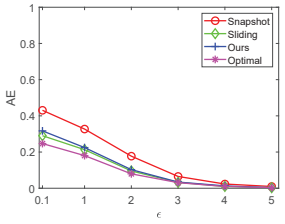


(a) AE vs. α

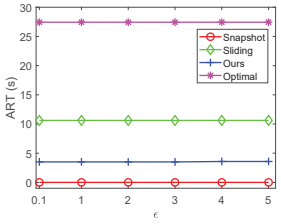


(b) ART vs. α

Fig. 9: AE and ART vs. α under LH.

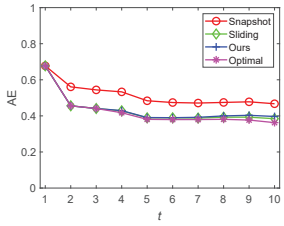


(a) AE vs. ϵ

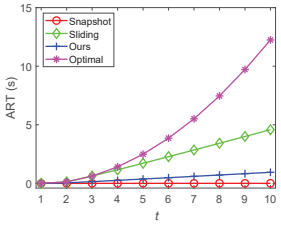


(b) ART vs. ϵ

Fig. 10: AE and ART vs. ϵ under Exp.

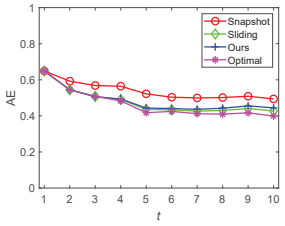


(a) AE vs. t

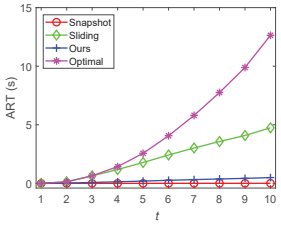


(b) ART vs. t

Fig. 11: AE and ART vs. t under LH over Gowalla dataset.



(a) AE vs. t



(b) ART vs. t

Fig. 12: AE and ART vs. t under Exp over Gowalla dataset.

privacy protection provided by Exp, the higher the inference accuracy. More importantly, the AE of Ours is always close to the ideal one achieved by Optimal, which confirms the effectiveness of Ours again. Fig. 9(b) and Fig. 10(b) show that the ARTs of different attack mechanisms are not affected by specific LPPM, which is anticipated. These results indicate that Ours could infer the user's true locations with high accuracy but low computation cost under different LPPMs.

D. Simulation Results on Real Dataset

Figs. 11 and 12 compare the AEs and ARTs of Snapshot, Sliding, Optimal, and Ours over the real dataset with trace length, t, varying from 1 to 10. We can see that as t increases, the AEs of four attack mechanisms all decrease but their ARTs increase, which is consistent with what we have observed from the synthetic datasets due to the same reason. Moreover, the AE of Ours is very close to that of Optimal, and the ART of Ours is only slightly higher than that under the Snapshot. The results on the real Gowalla dataset confirm the advantage of Ours over the other three attack mechanisms in achieving a better trade-off between inference accuracy and computation cost.

V. Related Work

Our work is most related to location inference attacks. Several inference attacks have been proposed in the literature. Shorkri *et al.* [15] introduced a Bayesian inference attack on an LPPM that estimates user's true location with the highest posterior probability. They also introduced another inference attack based on expected adversary error with respect to a distance function [3], [4]. Following their works, Theodorakopoulos *et al.* [16] and Yang *et al.* [17] studied a Stackelberg game between a user's defense and an adversary's inference attack which allows the adversary to obtain the optimal attack strategy. Niu *et al.* [18] introduced a long-term observation attack, which inferred user's location with all the received obfuscated locations generated from the same location to improve the inference accuracy. However, these attacks do not consider the temporal correlation among user's location, which could result in a low inference accuracy.

There are several inference attacks that consider the temporal correlation between the user's two adjacent locations. Shorkri *et al.* [15] modelled the user's mobility and moving behavior using a hidden Markov model and introduced a tracking attack to compute the distribution of user's trace based on the obfuscated trace. Oya *et al.* [9] also leveraged such

correlation to infer user's location by minimizing expected adversary error. Gambs *et al.* [19] predicted the next location of a user with the temporal correlation and the observed locations. Instead of using the Markov model, Ma *et al.* [20] leveraged the Conditional Random Fields to describe the temporal correlations. While they achieve higher inference accuracy, they incur high computation costs that grow exponentially as the number of obfuscated locations increases.

Since location inference attacks pose a serious threat to users' location privacy, many LPPMs [2]–[4], [20]–[24] have been proposed under the assumption that the adversary infers a user's location based on a single obfuscated location one at a time. Common to existing LPPMs is to perturb a user's true location to a noisy location used for LBS requests. For example, some LPPMs [1], [2], [21] perturbed a user's location to ensure geo-indistinguishability. As another example, several LPPMs [3], [4], [22] intended to maximize the adversary's estimation error under the Bayesian inference attack. In the continuous model, user frequently accesses an LBS [25], [26] and the reported locations usually exhibit temporal correlation, which can be exploited by the adversary to improve its inference accuracy [19], [27]. To defend such attacks, some works proposed new notion of location privacy [5], [6], [20], [23], [24], [28], [29] or adapted existing solutions designed for the sporadic model [7]–[9], [16], [30]. All these works are orthogonal to our work in this paper.

VI. Conclusions

In this paper, we have introduced a novel generalized inference attack for efficiently exploiting the temporal correlation among a user's adjacent location queries, which can strike a good balance between inference accuracy and computational complexity. Our simulation studies using both synthetic and real datasets have confirmed the advantages of the proposed attack over exiting attacks.

Acknowledgements

We would like to thank anonymous reviewers for their insightful comments that have helped improve the quality of this work. This work was supported in part by the US National Science Foundation under grants CNS-1933047 and CNS-1651954 (CAREER).

References

[1] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," in *CCS'13*, Berlin, Germany, Nov. 2013, pp. 901–914.

[2] K. Chatzikokolakis, E. ElSalamouny, and C. Palamidessi, "Efficient utility improvement for location privacy," *PoPETs*, vol. 2017, no. 4, pp. 308–328, Oct. 2017.

[3] R. Shokri, G. Theodorakopoulos, C. Troncoso, J.-P. Hubaux, and J.-Y. Le Boudec, "Protecting location privacy: Optimal strategy against localization attacks," in *CCS'12*, Raleigh, NC, Oct. 2012, pp. 617–627.

[4] R. Shokri, "Privacy games: Optimal user-centric data obfuscation," *PoPETs*, vol. 2015, no. 2, pp. 299–315, May 2015.

[5] Y. Xiao and L. Xiong, "Protecting locations with differential privacy under temporal correlations," in *CCS'15*, Denver, CO, Oct. 2015, pp. 1298–1309.

[6] Y. Xiao, L. Xiong, S. Zhang, and Y. Cao, "Loclok: Location cloaking with differential privacy via hidden markov model," *Proc. VLDB Endow.*, vol. 10, no. 12, pp. 1901–1904, Aug. 2017.

[7] K. Chatzikokolakis, C. Palamidessi, and M. Stronati, "A predictive differentially-private mechanism for mobility traces," *PoPETs*, pp. 21–41, July 2014.

[8] R. Al-Dhubhani and J. M. Cazalas, "An adaptive geo-indistinguishability mechanism for continuous lbs queries," *Wireless Networks*, vol. 24, no. 8, pp. 3221–3239, Nov. 2018.

[9] S. Oya, C. Troncoso, and F. Prez-Gonzlez, "Rethinking location privacy for unknown mobility behaviors," in *EuroS&P'19*, Stockholm, Sweden, June 2019, pp. 416–431.

[10] Y. He, J. Ni, L. T. Yang, W. Wei, X. Deng, D. Zou, and S. H. Ahmed, "Differentially private tripartite intelligent matching against inference attacks in ride-sharing services," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 22 583–22 595, 2022.

[11] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Wiley-Interscience, 2006.

[12] R. Trandafir, "Determination of a discrete distribution with given entropy," *Operational Research*, vol. 3, no. 1, pp. 41–46, January 2003.

[13] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: User movement in location-based social networks," in *KDD'11*, San Diego, CA, Aug. 2011, pp. 1082–1090.

[14] C. Dwork, "Differential privacy: A survey of results," in *TAMC'08*, Xi'an, China, April 2008, pp. 1–19.

[15] R. Shokri, G. Theodorakopoulos, J.-Y. L. Boudec, and J.-P. Hubaux, "Quantifying location privacy," in *S&P'11*, Berkeley, CA, May 2011, pp. 247–262.

[16] G. Theodorakopoulos, R. Shokri, C. Troncoso, J.-P. Hubaux, and J.-Y. Le Boudec, "Prolonging the hide-and-seek game: Optimal trajectory privacy for location-based services," in *WPES'14*, Scottsdale, AZ, Nov. 2014, pp. 73–82.

[17] X. Yang, L. Gao, J. Zheng, and W. Wei, "Location privacy preservation mechanism for location-based service with incomplete location data," *IEEE Access*, vol. 8, pp. 95 843–95 854, 2020.

[18] B. Niu, Y. Chen, Z. Wang, F. Li, B. Wang, and H. Li, "Eclipse: Preserving differential location privacy against long-term observation attacks," *IEEE TMC*, vol. 21, no. 1, pp. 125–138, 2022.

[19] S. Gambs, M.-O. Killijian, and M. N. n. del Prado Cortez, "Next place prediction using mobility Markov chains," in *MPM'12*, Bern, Switzerland, April 2012, pp. 1–6.

[20] Q. Ma, S. Zhang, T. Zhu, K. Liu, L. Zhang, W. He, and Y. Liu, "Plp: Protecting location privacy against correlation analyze attack in crowdsensing," *IEEE TMC*, vol. 16, no. 9, pp. 2588–2598, Sept. 2017.

[21] N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Optimal geo-indistinguishable mechanisms for location privacy," in *CCS'14*, Scottsdale, AZ, Nov. 2014, pp. 251–262.

[22] S. Oya, C. Troncoso, and F. Pérez-González, "Back to the drawing board: Revisiting the design of optimal location privacy-preserving mechanisms," in *CCS'17*, Dallas, TX, Oct. 2017, pp. 1959–1972.

[23] Y. Cao, M. Yoshikawa, Y. Xiao, and L. Xiong, "Quantifying differential privacy under temporal correlations," in *ICDE'17*, San Diego, CA, April 2017, pp. 821–832.

[24] ——, "Quantifying differential privacy in continuous data release under temporal correlations," *IEEE TKDE*, vol. 31, no. 7, pp. 1281–1295, July 2019.

[25] R. Mendes and J. a. Vilela, "On the effect of update frequency on geo-indistinguishability of mobility traces," in *WISEC'18*, Stockholm, Sweden, June 2018, pp. 271–276.

[26] R. Mendes, M. Cunha, and J. P. Vilela, "Impact of frequency of location reports on the privacy level of geo-indistinguishability," *PoPETs*, vol. 2020, no. 2, pp. 379–396, May 2020.

[27] S. Gambs, M.-O. Killijian, and M. N. n. del Prado Cortez, "Show me how you move and I will tell you who you are," in *SPRINGL'10*, San Jose, CA, Nov. 2010, pp. 34–41.

[28] Y. Cao, Y. Xiao, L. Xiong, and L. Bai, "Priste: From location privacy to spatiotemporal event privacy," in *ICDE'19*, Macao, April 2019, pp. 1606–1609.

[29] Y. Cao, Y. Xiao, L. Xiong, L. Bai, and M. Yoshikawa, "Priste: Protecting spatiotemporal event privacy in continuous location-based services," *Proc. VLDB Endow.*, vol. 12, no. 12, pp. 1866–1869, Aug. 2019.

[30] R. Shokri, G. Theodorakopoulos, and C. Troncoso, "Privacy games along location traces: A game-theoretic framework for optimizing location privacy," *ACM Trans. Priv. Secur.*, vol. 19, no. 4, pp. 1–31, Feb. 2017.