Online Location Trace Privacy: An Information Theoretic Approach

Wenjing Zhang, *Student Member, IEEE*, Ming Li, *Senior Member, IEEE*, Ravi Tandon, *Senior Member, IEEE*, and Hui Li, *Member, IEEE*

***Abstract*—We consider the problem of protecting individual user's location privacy at the trace-level and study the privacy-utility tradeoff, which has key applications in privacy-presreving Location-based Service (LBS). Existing works on Location Privacy Protection Mechanisms (LPPMs) have mainly focused on protecting single location, without taking into account the temporal correlations among locations within the trace, which can lead to higher privacy leakage when considering the whole trace. However, to date, there lacks a formal framework to quantify the trace-level location privacy leakage, and a practical mechanism to release location traces in an optimal and online manner. In this paper, we endeavor to solve this problem using an information-theoretic approach. We first propose a location trace privacy metric based on the mutual information between the original and released trace in an *offline* setting, and formulate the optimal location trace release problem that minimizes trace-level privacy leakage given a utility constraint. We also propose a privacy metric to capture trace-level privacy leakage in an *online* setting. As directly computing these metrics incur exponential complexity w.r.t. the trace length, we obtain upper and lower bounds on the trace-level privacy leakage by exploiting the Markov structure of the temporal location correlations, which are efficiently computable. The proposed upper bounds enable us to derive efficient online solutions (i.e., LPPMs) by modifying Blahut-Arimoto algorithm in rate-distortion theory. Then we validate the proposed upper and lower bounds and the actual leakage of our LPPM through extensive experiments over both synthetic and real-world location datasets. Our results show the superiority of our LPPM over existing LPPMs in terms of trace-level privacy-utility tradeoff, which is more conspicuous when the location trace is more correlated.**

***Index Terms*—Privacy metric, location trace privacy, temporal correlations, information-theoretic privacy, rate-distortion theory.**

I. Introduction

LOCATION-Based Service (LBS) has became an indispensable part of people's daily life [1], [2]. For instance, a user can take a picture and post it on Facebook with her current location; a student can find her friends by sharing locations through Foursquare; an Uber driver can locate the next passenger and search for the shortest path to a given destination using Google Map. Moreover, the significant amounts of data collected through LBS can be used in many other advanced applications, such as social relationship analysis, disease tracing, advertising, etc [3]–[5]. A location trace, which can be utilized in many applications, is a set of locations reported by a user while using LBS. For example, a user needs to periodically report her locations to a service provider in a navigation app (e.g., Google Map), which form a location trace and are highly correlated [6].

Privacy has been one of the most significant concerns in LBS applications. This is because the locations people share and report in LBS can be used to infer users' sensitive information, such as home addresses, travel plans, hospital visits, health conditions, etc., while the service providers cannot be fully trusted [7]. Many Location Privacy Protection Mechanisms (LPPMs) have been proposed to protect users' private locations against an untrusted LBS server [8]. For example, under a perturbation-based LPPM, a user's true location is distorted to a certain extent before being reported to a service provider. As a result, the user can still access LBS without sacrificing too much service quality (e.g., query accuracy), while the untrusted service provider cannot reveal her exact location.

A. Related work

Unfortunately, most of the existing LPPMs focus on protecting single location privacy [9]–[11], where locations reported by a user are not (or hardly) temporally correlated. Even though these approaches are practical and perfectly valid in single location scenario, they cannot be used to protect location trace privacy. These works are mainly based on the privacy definitions such as k-anonymity [12] and differential privacy [13], which are originally proposed to protect the existence of a single record in a database. Besides, k-anonymity has been disregarded as a reasonable privacy metric in [14]. In addition, cryptographic location privacy approaches apply encryption to protect user's locations [15], but those approaches are computational expensive even though they can provide strict privacy guarantee. There are also approaches based on spatial cloaking to protect location privacy [16], but spatial cloaking cannot be used as a privacy metric to evaluate LPPMs. Independently, authors in [11] propose to use the conditional entropy and the mutual information as complementary privacy metrics, and adopt Blahut-Arimoto algorithm to produce an LPPM that is almost optimal in terms of conditional entropy. Although it can be directly applied to a trace setting by releasing locations only depending on current location, it does not consider location correlations. Recent studies in [17], [18] have shown that, by

Wenjing Zhang is with the State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an, Shaanxi, 710071, China, and also with the Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ, 85721. Email: *xd.zhangwenjing@gmail.com*.

Ming Li and Ravi Tandon are with the Department of Electrical and Computer Engineering, University of Arizona, Tucson, AZ, 85721. Email: *lim@email.arizona.edu*; *tandonr@email.arizona.edu*.

Hui Li is with the State Key Laboratory of Integrated Service Networks, Xidian University, Xi'an, Shaanxi, 710071, China. Email: *lihui@mail.xidian.edu.cn*.

applying LPPMs for a single timestamp to a location trace, significant privacy leakage will be incurred due to temporal correlations within the trace. Moreover, reconstructing user's traces from obfuscated individual locations is also possible [19], [20]. Although a couple of works based on game theory [21], [22] or extended notions of differential privacy [17] have been proposed to take into account the temporal correlations of locations, and there is also some work in which authors measure the trace-level privacy by averaging individual location privacy [23], there still lacks proper privacy metrics that take the location correlations inside a trace into account when quantifying the privacy leakage of an entire location trace.

Even though there are some related works on privacy metrics [24]–[30] used to quantify information leakage based on information theory, we argue that these privacy metrics are either not applicable or not practical when used on location trace privacy. Ma et al. [24] proposed a privacy metric for time-series data to quantify the amount of information available to the adversary when he tries to infer the original data given any range of released data. However, this metric quantifies the privacy leakage about a single timestamp's data point rather than the entire time-series. Besides, they consider an offline setting instead of an online setting. Shokri et al. in [25], [26] proposed privacy metrics that quantify attacker's location estimation error under specific types of inference attacks, and these metrics take inherently location correlations into account. Even so, we believe that the information-theoretic metrics proposed in our work which also consider correlations into account are still useful, because they provide another point of view that complements the average adversary error privacy metric. Cuff et al. [27] proposed a metric based on conditional mutual information to interpret differential privacy. However, the privacy guarantee of this metric is too strong to achieve for location trace privacy. This is because the adversary in this setting is assumed to know all the other locations in a location trace except for one location; since locations are correlated, much noise has to be added to protect a single location in a location trace and thus leads to little utility. Theoretically speaking, the privacy metric proposed in [28] could be applied to location trace privacy. However, deriving the optimal LPPM from their metric is impractical, since their optimal mechanism is based on the achievability of rate distortion, which uses data compression and joint typical decoding on a large domain size. The privacy metrics proposed in [29], [30] also face scalability issues w.r.t. trace length, so they are not practical when applied to quantify trace privacy.

B. Contributions

In this paper, we propose a novel location trace privacy metric to quantify the information leakage between the original and the released trace when any LPPM is adopted. Our privacy metric helps to further understand the information leakage of different LPPMs in practice, and offers a formal way of comparing privacy levels achieved by existing and future LPPMs. Besides, we derive the optimal LPPM by formulating the optimal location trace release problem as a minimization problem over trace-level privacy leakage given a utility constraint. In addition, we address practical challenges encountered when directly computing this metric. The major contributions of this paper are summarized as follows:

- We propose privacy metrics to quantify trace-level information leakage both in *offline* and *online* setting. By leveraging the mutual information in information theory, we formulate the optimization problem that minimizes trace-level privacy leakage given a utility constraint to derive the *optimal* location trace release mechanism in the *online* setting, which is more interesting and practical. Our metric is generic and independent of any specific inference attack. The motivation for choosing mutual information as the privacy metric comes from the fact that priors and correlations naturally exist in location traces, and we need a privacy metric to capture the priors and correlations in location traces in a principle and clear way.
- We address a practical challenge encountered when solving the above optimization problem in *online* setting. Since directly computing the privacy metric leads to exponential complexity with respect to the trace length, we derive upper and lower bounds of the privacy leakage by exploiting the Markov structure of the temporal location correlations, which are efficiently computable. The proposed upper bound enables us to derive efficient online solutions (LPPMs) by modifying the Blahut-Arimoto algorithm [31] in rate-distortion theory. In particular, our LPPM can be pre-computed in advance and then used for online location release with very high efficiency.
- Using our privacy metric, we compare our LPPM with two state-of-the-art LPPMs via extensive experiments over both synthetic and real-world location datasets. Our results demonstrate that our LPPM reveals the least amount of information under the same utility constraint. Moreover, its advantage in the privacy-utility tradeoff becomes even greater when location traces become more correlated. We also show the efficiency of our LPPM, where the offline pre-computation requires a reasonable time, and the online release is very fast.

The rest of this paper is organized as follows. We present the problem statement in Section II, and provide the preliminaries in Section III. Section IV describes the main results for the online privacy-utility tradeoff for location traces together with algorithms for generating optimal LPPMs based on upper bounds in Section V. Section VI presents experimental results, followed by conclusion and future work in Section VII. Finally, the proofs for all the theoretical results are presented in Appendix.

II. Problem Statement

In this section, we present the problem setting, describe the threat model, define the privacy and utility metrics for a location trace, and formally present our problem. The notations introduced throughout the paper are summarized in Table I.

A. Location Trace Model

We represent user's location L_i at timestamp i, as a triplet (x_i, y_i, i), where x_i, y_i, and i represent the latitude coordinate, longitude coordinate and timestamp respectively. A location

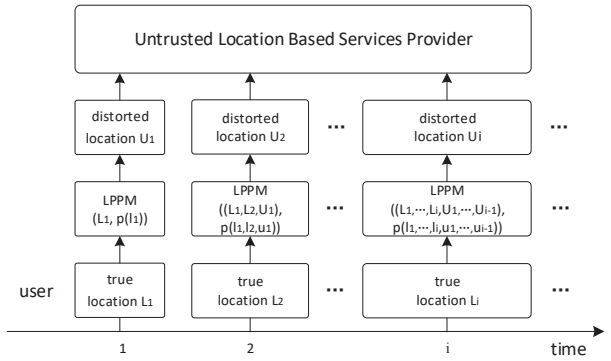


Fig. 1. Problem Setting: Online Privacy-preserving Location Release

trace $\boldsymbol{L}$ with length T is represented by a sequence of locations $(L_1, L_2, ..., L_T)$. The timestamp i and length T take integer values. Furthermore, we assume that the user moves within N discrete locations.

In our model, we assume user's location traces are generated from an underlying probability distribution, which can be obtained from user's initial location probability distribution and a mobility model. Specifically, we denote user's initial location probability distribution by an N-length vector p_1, and consider user's mobility model as a first-order Markov model denoted by a Markov transition matrix $\boldsymbol{M}$ [17], [25], [32]. Each element in p_1 represents the probability that a user stays in a certain location. In addition, we use $p_{m,n}$ to denote an element at the mth row and nth column in $\boldsymbol{M}$, i.e., $p_{m,n}$ represents the probability that a user moves from location m to location n. Then the probability distribution of user's location at timestamp i is $p_i = p_1\boldsymbol{M^{i-1}}$. Moreover, the probability distribution of a location trace with length i is the joint distribution of all the locations up to timestamp i, i.e., $p(l_1, l_2, ..., l_i)$, which can be obtained from user's initial location probability distribution p_1 and the Markov transition matrix $\boldsymbol{M}$. In particular, the practicality of using the Markov mobility model is discussed in [25].

B. Online Privacy-preserving Location Release

We consider the problem setting illustrated in Fig. 1, where a privacy-conscious user releases a distorted location U_i instead of her true location L_i to the untrusted service provider at each timestamp i to obtain services. Specifically, the user releases her locations in an *online manner*, which means that at timestamp i, the user generates and then releases a distorted location U_i according to the joint probability distribution of all her true locations available up to timestamp i and the past distorted locations, i.e., $p(l_1, ..., l_i, u_1, ..., u_{i-1})$. This can ensure that the LPPM used at each timestamp i takes the temporal correlations among a location trace into account. We assume $\mathcal{L}_i = \mathcal{U}_i$ for simplicity, i.e., the alphabet of true locations and released locations are the same.

C. Threat Model

As we can see from Fig. 1, after a certain time period $1, 2, ..., T$, the untrusted service provider can observe the released (distorted) location trace in the form of $(U_1, U_2, ..., U_T)$. Then the service provider could infer user's private location information based on the released location trace $(U_1, U_2, ..., U_T)$. We assume that the untrusted service provider has full statistical knowledge of user's locations, i.e., user's initial location probability distribution and her mobility model. Furthermore, we do not make any restriction on the computational capability of the untrusted service provider. In principle, it can use this statistical knowledge and the released location trace to launch any type of inference attack. Now our goal is to understand the fundamental information leakage (i.e., privacy leakage) arising from releasing user's distorted location trace in such scenarios.

D. Privacy and Utility Metrics for a Location Trace

We use random variables L and U to represent user's true location and released location respectively, and their lower case l and u are possible values of these two random variables. Random vectors $\boldsymbol{L} = (L_1, L_2, ..., L_T)$ and $\boldsymbol{U} = (U_1, U_2, ..., U_T)$ represent user's true location trace with length T and the released location trace with length T respectively, and the lower case $\boldsymbol{l}$ and $\boldsymbol{u}$ are possible values of these two random vectors.

***Definition** 1:* **Privacy Metric for a location trace.** *For a certain time period $1, 2, ..., T$, given user's true location trace $\boldsymbol{L} = (L_1, L_2, ..., L_T)$ and her released location trace $\boldsymbol{U} = (U_1, U_2, ..., U_T)$, the information leakage introduced by the released location trace is defined as $I(\boldsymbol{L}; \boldsymbol{U}) = I(L_1, L_2, ..., L_T; U_1, U_2, ..., U_T)$, where $I(\boldsymbol{L}; \boldsymbol{U})$ is the mutual information between user's true location trace and the distorted location trace she releases to the untrusted service provider. We use $I(\boldsymbol{L}; \boldsymbol{U})$ as the privacy metric for a location trace.*

***Remark** 1: In the case that the untrusted service provider has additional background knowledge (e.g., user's social network information such as her co-locations) other than user's initial location probability distribution and her mobility model, our privacy metric for a location trace can be generalized as the metric $I(\boldsymbol{L}; \boldsymbol{U}, Z)$ to take the additional background knowledge into account, where Z is a random variable representing the additional background knowledge.*

This general definition measures the fundamental information leakage of user's true location trace introduced by the released trace. By limiting the information leakage for a location trace to a certain level, an LPPM generated based on our privacy metric provides the privacy guarantee that the less information leakage introduced by releasing the distorted trace, the higher privacy level can be preserved for the user.

We also want to highlight that mutual information and conditional entropy are two alternative privacy metrics in our problem. This is because when the prior distribution of a location trace $\boldsymbol{L}$ (i.e., the entropy $H(\boldsymbol{L})$) is fixed, knowing that $I(\boldsymbol{L}; \boldsymbol{U}) = H(\boldsymbol{L}) - H(\boldsymbol{L}|\boldsymbol{U})$, we can conclude that the less the mutual information $I(\boldsymbol{L}; \boldsymbol{U})$ is, the larger the conditional entropy $H(\boldsymbol{L}|\boldsymbol{U})$ will be. In other words, based on $H(\boldsymbol{L})$ and $I(\boldsymbol{L}; \boldsymbol{U})$, the conditional entropy $H(\boldsymbol{L}|\boldsymbol{U})$ can be easily calculated. Since these two metrics are alternative in our problem, we only use mutual information as the privacy metric in this paper.

TABLE I
NOTATIONS

Symbol	Description
i, T	Timestamp, length of a location trace
$\boldsymbol{L} \in \boldsymbol{\mathcal{L}}, \boldsymbol{U} \in \boldsymbol{\mathcal{U}}$	Random vectors representing the true and released location trace with length T
$L_i \in \mathcal{L}_i, U_i \in \mathcal{U}_i$	Random variables representing the true and released location at timestamp i
$\boldsymbol{L}^i$, $\boldsymbol{U}^{i-1}$	Random vectors representing $(L_1, ..., L_i)$, and $(U_1, ..., U_{i-1})$
$\boldsymbol{l}, \boldsymbol{u}, l_i, u_i, \boldsymbol{l}^i, \boldsymbol{u}^{i-1}$	Possible values of $\boldsymbol{L}$, $\boldsymbol{U}$, L_i, U_i, $\boldsymbol{L}^i$, $\boldsymbol{U}^{i-1}$
$p(\cdot), r(\cdot)$	Probability distribution of the true and released location
$q(\cdot\,\vert\,\cdot)$, $p(\cdot, \cdot)$	Conditional, joint probability distribution
$\boldsymbol{M}$	Markov transition matrix

However, in order to obtain utility from LBS, the distortion introduced by the released location trace should be limited to a certain threshold. Hence, in order to capture the utility of an LPPM, we define the following utility metric.

***Definition** 2:* **Utility Metric for a Location Trace.** *For a certain time period $1, 2, ..., T$, given user's true location trace $\boldsymbol{L} = (L_1, L_2, ..., L_T)$ and released location trace $\boldsymbol{U} = (U_1, U_2, ..., U_T)$, the utility metric for a location trace is defined as $D(\boldsymbol{L}; \boldsymbol{U}) = \sum_{i=1}^{T} D(L_i; U_i)$, where $D(L_i; U_i)$ is the expected distortion for the released location at timestamp i (i.e., U_i) and defined as $D(L_i; U_i) = \sum_{l_i, u_i} p(l_i) q(u_i|l_i) d(l_i, u_i)$, where $d(l_i, u_i)$ is the distortion function (e.g., Hamming distance or Euclidean distance). The utility (distortion) constraint for the released location at timestamp i is defined as $D(L_i; U_i) \leq D_i, i = 1, 2, ..., T$, where D_i is the distortion assigned to the released location at timestamp i in a location trace, which implies that the total distortion for a location trace $D \leq \sum_{i=1}^{T} D_i$.*

The definition of utility metric implies that the total distortion for a location trace actually depends on the individual distortion of the released location at each timestamp. This is reasonable since the user obtains utility from LBS in an *online manner*, thus the utility for the released location at each timestamp should be ensured by an individual utility constraint, which could be different from one another due to the type of LBS accessed by the user at a specific timestamp.

E. Problem

Intuitively, the less information leakage required by the user, the less utility the user can get, and vice versa. Therefore, there exists a privacy-utility tradeoff when designing LPPMs based on our privacy metric. A natural question arises as what is the minimum information leakage subject to a utility constraint from an information-theoretic perspective and how to design an LPPM to achieve this minimum information leakage. We formulate this problem in the following proposition.

***Proposition** 1:* **Offline Privacy-Utility tradeoff for Location Traces.** *For a certain time period $1, 2, ..., T$, given user's true location trace $\boldsymbol{L} = (L_1, L_2, ..., L_T)$, her released location trace $\boldsymbol{U} = (U_1, U_2, ..., U_T)$, and the utility constraint $D \leq \sum_{i=1}^{T} D_i$, an LPPM $q(\boldsymbol{u}|\boldsymbol{l})$ is to say achieving the minimum information leakage of a location trace subject to the utility constraint D when it is the solution of the following optimization problem:*

$$\mathcal{L}^*_{\text{offline}}(D) = \min_{q(\boldsymbol{u}|\boldsymbol{l}): \{D(L_i;U_i) \leq D_i\}_{i=1}^{T}} I(\boldsymbol{L}; \boldsymbol{U}),$$

where $I(\boldsymbol{L}; \boldsymbol{U})$ is the privacy metric for a location trace.

Proposition 1 provides a general framework for the *offline*[1] privacy-preserving location trace release, which contains two parts, i.e., a pre-computing process for generating an optimal LPPM $q^*(\boldsymbol{u}|\boldsymbol{l})$ and a location trace releasing process according to this LPPM. Specifically, once the optimal LPPM $q^*(\boldsymbol{u}|\boldsymbol{l})$ is obtained, it works in an *offline* manner: given user's true location trace as $\boldsymbol{L}$, the user will sample from $q^*(\boldsymbol{u}|\boldsymbol{l})$ to obtain the distorted location trace $\boldsymbol{U}$ that achieves the minimum information leakage subject to the utility constraint D, and then release $\boldsymbol{U}$ to the service provider.

As we will show in Section IV, although this *offline* privacy-utility tradeoff for location traces is theoretically meaningful, it is actually extremely hard to find in practice. To this end, we will introduce the problem of *online* privacy-utility tradeoff for location traces and the methodology of analyzing and characterizing this tradeoff in the rest of the paper.

III. PRELIMINARIES

In this section, we present the background for the rate-distortion function and the algorithms used for its calculation. The rate distortion problem has been considered for the problem of lossy compression, where the goal is to minimize the compression rate subject to a distortion constraint. We notice that there is a close connection between the privacy-utility tradeoff in Proposition 1 and the rate distortion problem. This connection has also been studied in [28], [29]. Specifically, they view the rate and distortion as analogous to the information leakage and utility respectively when analyzing the privacy-utility tradeoff. However, these works are looking at using the principle of rate distortion function on privacy-utility tradeoff in the setting of databases. Even though the connection between rate distortion and the privacy-utility trade-off has also been studied for individual locations in [11], it has not been studied for the setting of location trace privacy. Moreover, if we keep using the same principle of this connection to study the privacy-utility problem for location traces, there are practical challenges that arise in designing efficient mechanisms as we will mention in Section IV. In the following, we briefly present the rate distortion problem, it's computation and connection to Proposition 1.

***Definition** 3:* **Rate Distortion Function [33].** *If the input of an encoder is X and the output of the corresponding decoder is $\hat{X}$, the rate distortion function $R(D)$ for a source $X \sim p(x)$ with distortion measure $d(x, \hat{x})$ is defined as*

$$\begin{aligned} R(D) &= \min_{\substack{p(\hat{x}|x): \\ \sum_{x,\hat{x}} p(x)p(\hat{x}|x)d(x,\hat{x}) \leq D}} I(X; \hat{X}) \\ &= \min_{\substack{p(\hat{x}|x): \\ \sum_{x,\hat{x}} p(x)p(\hat{x}|x)d(x,\hat{x}) \leq D}} p(x)p(\hat{x}|x) \log \frac{p(\hat{x}|x)}{p(\hat{x})}, \end{aligned} \tag{1}$$

[1] In contrast to the *online setting* where a user releases her distorted location individually at each timestamp, in the *offline setting*, the user releases her entire distorted location trace to the service provider once for all.

where the minimization is over all conditional distributions $q(\hat{x}|x)$ for which the joint distribution $p(x,\hat{x}) = p(x)p(\hat{x}|x)$ satisfies the expected distortion constraint.

We next describe a general algorithm for finding the minimum distance between two convex sets, and this algorithm can be in turn used to find the solution to the optimization problem for the rate distortion function.

General Algorithm for Finding the Minimum Distance between Two Convex Sets [33], [34]. Given two convex sets A and B, the minimum distance between them $d_{min} = \min_{a\in A}\min_{b\in B} d(a,b)$, where $d(a,b)$ is the Euclidean distance between a and b, can be found by the following steps: first we can take any point $x \in A$, and find the $y \in B$ that is closest to it. Then fix this y and find its closest point in A. Iterative applications of this process decreases the distance at each step. The result in [34] has shown that if the sets are convex and if the distance satisfies certain conditions, this alternating minimization algorithm will indeed converge to the minimum. In particular, if the sets are sets of probability distributions and the distance measure is the relative entropy, the algorithm does converge to the minimum relative entropy between the two sets of distributions.

Last, we briefly review the technical details of the Blahut-Arimoto algorithm which utilizes the above idea to compute the rate distortion function.

Blahut-Arimoto algorithm for Computing the Rate Distortion Function. Blahut-Arimoto algorithm [31], [33] is an iterative algorithm that eventually converges to the optimal solution of the convex optimization problem in the rate distortion function. Specifically, in this algorithm, it first chooses an initial distribution for $r(\hat{x})$ (e.g., a uniform distribution), then uses $r(\hat{x})$ to compute $q(\hat{x}|x) = \frac{r(\hat{x})e^{-\lambda d(x,\hat{x})}}{\sum_{\hat{x}} r(\hat{x})e^{-\lambda d(x,\hat{x})}}$. After obtaining $q(\hat{x}|x)$, it updates $r(\hat{x})$ by setting $r(\hat{x}) = \sum_x p(x)q(\hat{x}|x)$. Then it uses $r(\hat{x})$ to update $q(\hat{x}|x)$ by setting $q(\hat{x}|x) = \frac{r(\hat{x})e^{-\lambda d(x,\hat{x})}}{\sum_{\hat{x}} r(\hat{x})e^{-\lambda d(x,\hat{x})}}$. The optimal solution $q(\hat{x}|x)$ that minimizes the rate distortion function can be obtained by repeating the above iteration between $r(\hat{x})$ and $q(\hat{x}|x)$ until convergence.

In principle, the Blahut-Arimoto algorithm can be used to compute the optimal LPPM $q^*(\boldsymbol{u}|\boldsymbol{l})$ in Proposition 1. However, as we will discuss in the next section, there are significant practical challenges when directly using the Blahut-Arimoto algorithm on our problem.

IV. Online Privacy-Utility tradeoff for Location Traces

A. Practical Challenge in Finding the Optimal Offline Privacy-Utility Tradeoff for Location Traces

Directly using the Blahut-Arimoto algorithm on the optimization problem in Proposition 1 incurs exponential complexity. This is because we need to characterize the values of $q(\boldsymbol{u}|\boldsymbol{l})$ for all possible combinations $(\boldsymbol{u},\boldsymbol{l}) \in \boldsymbol{\mathcal{L}} \times \boldsymbol{\mathcal{U}}$. In other words, we have to solve the optimization problem over $|\boldsymbol{\mathcal{U}}||\boldsymbol{\mathcal{L}}|$ variables in order to find the optimal solution $q(\boldsymbol{u}|\boldsymbol{l})$. Specifically, if we consider a user moving within N locations and her distorted locations are also taken from these N locations, then the number of variables will be N^{2T} when the user wants to release a location trace with length T, since we have $|\boldsymbol{\mathcal{U}}| = |\boldsymbol{\mathcal{L}}| = N^T$ in this case. As we can see, with an increase in the length T of a location trace, the number of variables increases exponentially. In addition to this computation complexity issue, the problem of the *offline* privacy-utility tradeoff also does not tell us about the information leakage for the online privacy-preserving location release, i.e., it does not capture the online nature of this setting.

Therefore, in the following, we propose a new problem, i.e., the *online* privacy-utility tradeoff, to analyze the minimum information leakage subject to a certain utility constraint in the online privacy-preserving location release setting. Even though the computation complexity issue still remains in finding the optimal *online* privacy-utility tradeoff, we will show how to address this issue by deriving upper and lower bounds on the tradeoff which are efficiently computable. Interestingly, these upper and lower bounds derived for the *online* privacy-utility tradeoff also provide us an insight to understand and analyze the *offline* privacy-utility tradeoff.

B. Privacy-Utility Tradeoff for Online Location Release Mechanisms

We first introduce the definition of the privacy leakage for online location release mechanisms as below,

***Definition** 4:* **Privacy Leakage for Online Location Release Mechanisms.** *For a certain time period $1, 2, ..., T$, when a user is releasing her locations in an online manner (i.e., she sequentially releases her locations which form a released location trace), the actual privacy leakage introduced in this online location release setting is defined as*

$$\mathcal{L}^{\textit{Actual}}_{\textit{online}}(\textit{LPPM}) = \sum_{i=1}^{T} I(\boldsymbol{L}^i; U_i|\boldsymbol{U}^{i-1}), \tag{2}$$

where the LPPM could be generated based on any type of approaches, and $\boldsymbol{L}^i$, U_i, $\boldsymbol{U}^{i-1}$ are described in Table I. We use $\mathcal{L}^{\textit{Actual}}_{\textit{online}}(\textit{LPPM})$ as the privacy metric to evaluate the actual privacy leakage for online location release mechanisms.

We will describe how to evaluate the actual privacy leakage $\mathcal{L}^{\text{Actual}}_{\text{online}}(\text{LPPM})$ for specific LPPMs in detail in Section V-C.

Next, we present the problem of *online* privacy-utility tradeoff for location traces in the following proposition.

***Proposition** 2:* **Online Privacy-Utility tradeoff for Location Traces.** *The tradeoff between privacy leakage and distortion for online release mechanisms is given as follows,*

$$\mathcal{L}^*_{\text{online}}(D) = \sum_{i=1}^{T} \min_{\substack{q(u_i|\boldsymbol{l}^i,\boldsymbol{u}^{i-1}):\\ D(L_i;U_i)\le D_i}} I(\boldsymbol{L}^i; U_i|\boldsymbol{U}^{i-1}), \tag{3}$$

where D_i represents the distortion assigned to the ith optimization problem in the summation in (3), $\boldsymbol{L}^i$, $\boldsymbol{U}^{i-1}$, $\boldsymbol{l}^i$ and $\boldsymbol{u}^{i-1}$ are described in Table I. Furthermore, the online privacy-utility tradeoff is always greater or equal to the offline privacy-utility tradeoff, i.e., $L^*_{\textit{offline}}(D) \le L^*_{\textit{online}}(D)$.

The proof of $L^*_{\text{offline}}(D) \le L^*_{\text{online}}(D)$ is given in Appendix.

In Proposition 2, we can see that finding the *online* privacy-utility tradeoff requires solving the optimization problems in (3) individually where the objective functions are in the form of $I(\boldsymbol{L}^i; U_i|\boldsymbol{U}^{i-1})$. In addition, by leveraging this online

location release mechanism, a user can release her location u_i according to the LPPM $q(u_i|\boldsymbol{l}^i, \boldsymbol{u}^{i-1})$ at each timestamp i, instead of releasing her whole location trace $\boldsymbol{u}$ according to the LPPM $q(\boldsymbol{u}|\boldsymbol{l})$ obtained in Proposition 1 in the offline setting. Specifically, the optimization problems in (3) are solved in a sequential manner from timestamp 1 to timestamp T. In particular, the LPPM $q(u_i|\boldsymbol{l}^i, \boldsymbol{u}^{i-1})$ obtained at timestamp i (i.e., the solution in the ith optimization problem in (3)) will be used as an input for the problem at timestamp $i+1$ to find the optimal LPPM $q(u_{i+1}|\boldsymbol{l}^{i+1}, \boldsymbol{u}^i)$ that minimize the $i+1$th optimization problem in (3).

However, we notice that we still need to characterize the value of $q(u_i|\boldsymbol{l}^i, \boldsymbol{u}^{i-1})$ for all possible combinations $(\boldsymbol{u}^i, \boldsymbol{l}^i) \in \boldsymbol{\mathcal{L}^i} \times \boldsymbol{\mathcal{U}^i}$, i.e., the length of a location trace is still involved in the objective function $I(\boldsymbol{L}^i; U_i|\boldsymbol{U}^{i-1})$ even though we are considering the online location release setting. The essential reason is that as long as the number of variables in the objective function depends on the length of a location trace, the exponential complexity still exists. If we want to remove the effect caused by the length of a location trace in the optimization problem and make the computation for the optimal solution more efficient, the number of variables in $I(\boldsymbol{L}^i; U_i|\boldsymbol{U}^{i-1})$ needs to be independent of the length of a location trace. To achieve this goal, we obtain upper and lower bounds on $L^*_{\text{online}}(D)$ by relaxing the objective function, which leads to the numbers of variables in the new objective functions in the upper and lower bounds become independent of the length of a location trace.

In the following, we present the main results for *online* privacy-utility tradeoff for location traces.

1) Upper and Lower Bounds on Online Privacy-Utility Tradeoff with Markov Release Restriction: Intuitively, a location is more likely correlated with those that are closer to it in terms of time span. Based on this intuition, we make a Markov restriction on the location release mechanisms that the released location U_i at timestamp i only depends on the current true location L_i, the previous released location U_{i-1}, and the previous true location L_{i-1}. We want to highlight that our system model allows for potentially more complex mechanisms but for complexity issues we focus on Markov release. The location release mechanism with Markov restriction is shown in Fig. 2.

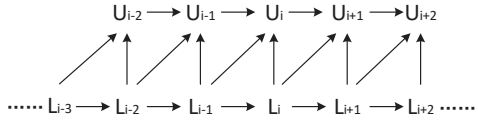


Fig. 2. Location release mechanism with Markov restriction

Based on this restriction on the location release mechanisms, we derive the upper and lower bounds on $\mathcal{L}^*_{\text{online}}(D)$ and present them in the following theorem.

***Theorem** 1:* **Upper and Lower Bounds on online Privacy-Utility Tradeoff with Markov Release Restriction.** *With the Markov restriction on the location release mechanism, the optimal online privacy-utility tradeoff $\mathcal{L}^*_{online}(D)$ can be upper and lower bounded as follows:*

$$\mathcal{L}^{\text{Markov}}_{\text{lower}}(D) \leq \mathcal{L}^*_{\text{online}}(D) \leq \mathcal{L}^{\text{Markov}}_{\text{upper}}(D), \tag{4}$$

where

$$\mathcal{L}^{\text{Markov}}_{\text{upper}}(D) = \sum_{i=1}^{T} \min_{\substack{q(u_i|l_i,u_{i-1},l_{i-1}):\\ D(L_i;U_i)\leq D_i}} I(L_i, L_{i-1}; U_i|U_{i-1}), \tag{5}$$

$$\mathcal{L}^{\text{Markov}}_{\text{lower}}(D) = \sum_{i=1}^{T} \min_{\substack{q(u_i|l_i,u_{i-1},l_{i-1}):\\ D(L_i;U_i)\leq D_i}} I(L_i; U_i|U_{i-1}, L_{i-1}). \tag{6}$$

The proof of Theorem 1 is given in Appendix.

We want to highlight that we will use $\mathcal{L}^{\text{Markov}}_{\text{upper}}(D)$ to generate LPPMs rather than $\mathcal{L}^{\text{Markov}}_{\text{lower}}(D)$, since we usually care more about how much information at most will be leaked when designing privacy-preserving mechanisms.

***Definition** 5: The optimal LPPM for a T-length location trace, generated based on $\mathcal{L}^{Markov}_{upper}(D)$ from timestamp 1 to T in a online manner, is defined as LPPM* = $\{LPPM^*_1,...,LPPM^*_i,...,LPPM^*_T\}$, where $LPPM^*_i$ is the optimal solution $q(u_i|l_i, u_{i-1}, l_{i-1})$ obtained from solving the ith optimization problem in the summation in (5), i.e.,*

$$LPPM^*_i = \arg \min_{\substack{q(u_i|l_i,u_{i-1},l_{i-1}):\\ D(L_i;U_i)\leq D_i}} I(L_i, L_{i-1}; U_i|U_{i-1}). \tag{7}$$

***Corollary** 1: The actual leakage of LPPM* evaluated by $\mathcal{L}^{Actual}_{online}(LPPM)$ is sandwiched between $\mathcal{L}^{Markov}_{lower}(D)$ and $\mathcal{L}^{Markov}_{upper}(D)$, i.e.,*

$$\mathcal{L}^{\text{Markov}}_{\text{lower}}(D) \leq \mathcal{L}^{\text{Actual}}_{\text{online}}(\text{LPPM}^*) \leq \mathcal{L}^{\text{Markov}}_{\text{upper}}(D). \tag{8}$$

The proof of Corollary 1 is presented in Appendix.

Corollary 1 provides us the privacy guarantee that the exact information leakage for an entire location trace released by LPPM* sequentially from timestamp 1 to T is sandwiched between $\mathcal{L}^{\text{Markov}}_{\text{lower}}(D)$ and $\mathcal{L}^{\text{Markov}}_{\text{upper}}(D)$. Therefore, even though directly computing the exact information leakage for an entire location trace is computationally challenging, we can still know the approximate information leakage by sandwiching the exact information leakage between $\mathcal{L}^{\text{Markov}}_{\text{lower}}(D)$ and $\mathcal{L}^{\text{Markov}}_{\text{upper}}(D)$. Besides, Corollary 1 strengthens our statement again that we should generate LPPM according to $\mathcal{L}^{\text{Markov}}_{\text{upper}}(D)$ rather than $\mathcal{L}^{\text{Markov}}_{\text{lower}}(D)$, since the actual leakage for an entire location trace is upper bounded by $L^{\text{Markov}}_{\text{upper}}(D)$. Here we need to mention that Theorem 1 and Corollary 1 are only valid under the Markov release restriction.

***Corollary** 2: Given the LPPM^{cl} for a T-length location trace that releases locations only depending on the current location at each timestamp, i.e., the LPPM at timestamp i is in the form of $q(u_i|l_i)$, where "cl" is shorted for current location, the actual leakage of LPPM^{cl} evaluated by $\mathcal{L}^{Actual}_{online}(LPPM)$ is upper and lower bounded as below,*

$$I_{\text{lower}} \leq \mathcal{L}^{\text{Actual}}_{\text{online}}(\text{LPPM}^{\text{cl}}) \leq I_{\text{upper}}, \tag{9}$$

where $I_{\text{upper}} = \sum_{i=1}^{T} I(L_i, L_{i-1}; U_i|U_{i-1})$, and $I_{\text{lower}} = \sum_{i=1}^{T} I(L_i; U_i|U_{i-1}, L_{i-1})$.

Since LPPMs that release locations only depending on the current location also satisfy the Markov release restriction, Corollary 2 can be easily proved based on the proof of Corollary 1. Thus we omit its proof for space limitation.

We want to mention that the previous LPPMs designed for single location scenario (e.g., [10], [11]) can be straightforwardly applied to location traces by releasing locations only depending on the current location. One take away from Corollary 2 is that it provides us the upper and lower bounds on the actual online leakage for a location trace when LPPMs release locations only depending on the current location. Therefore, calculating I_{upper} and I_{lower} is also meaningful for this type of LPPMs, especially when their actual leakage is difficult to calculate due to the exponential computation complexity, which we will explain in detail in Section V-C.

Now we briefly analyze the generating process for LPPM*. As we can see from Theorem 1, this process is quite efficient, since we only need to characterize the values of $q(u_i|l_i, u_{i-1}, l_{i-1})$ for N^4 possible combinations in $\mathcal{L}^{\text{Markov}}_{\text{upper}}(D)$, which will make the computation much more efficient compared with characterizing $q(\boldsymbol{u}|\boldsymbol{l})$ for N^{2T} possible combinations in $\mathcal{L}^*_{\text{online}}(D)$. Moreover, Theorem 1 provides a framework for privacy-preserving location trace release under the Markov release restriction . This framework is similar in spirit to the one in Proposition 2, since both of them contain the pre-computation part for generating LPPM and location trace release part according to this LPPM. In terms of how to use this LPPM to release locations in an online manner, the user follows the following procedure: after she obtains LPPM$^*_i$ at timestamp i, she can sample from the conditional probability distribution $q(u_i|l_i, u_{i-1}, l_{i-1})$ based on her current true location L_i, the previous released and true locations U_{i-1} and L_{i-1} to obtain the released location U_i. Next, in order to obtain the LPPM$^*_{i+1}$ $q(u_{i+1}|l_{i+1}, u_i, l_i)$ to release her location U_{i+1} at the next timestamp $i+1$, the user will input her current LPPM$^*_i$ $q(u_i|l_i, u_{i-1}, l_{i-1})$ to the $i+1$th optimization problem in (5), which we will explain in detail in Section V.

As we can see from Theorem 1, we address the computation challenge by deriving the upper and lower bounds on the *online* privacy-utility tradeoff by making a Markov restriction on location release mechanism. To make our analysis more general, we also prove the upper and lower bounds without making this restriction and the result is shown as below.

2) Upper and Lower Bounds on Online Privacy-Utility Tradeoff for Generic Online Location Release Mechanisms: In the case of generic online location release, i.e., when we do not make the Markov restriction on the release mechanism, the result is shown in the following theorem.

***Theorem** 2:* **Upper and Lower Bounds on online Privacy-Utility Tradeoff for Generic Online Location Release Mechanisms.** *For generic online LPPMs, the optimal online privacy-utility tradeoff $\mathcal{L}^*_{online}(D)$ can be upper and lower bounded as follows:*

$$\mathcal{L}_{\text{lower}}(D) \leq \mathcal{L}^*_{\text{offline}}(D) \leq \mathcal{L}^*_{\text{online}}(D) \leq \mathcal{L}_{\text{upper}}(D), \quad (10)$$

where

$$\mathcal{L}_{\text{upper}}(D) = \sum_{i=1}^{T} \min_{\substack{q(u_i|l_i):\\ D(L_i;U_i)\leq D_i}} I(L_i;U_i), \quad (11)$$

$$\mathcal{L}_{\text{lower}}(D) = T \cdot \min_i \min_{\substack{q(u_i|l_i):\\ D(L_i;U_i)\leq D_i}} I(L_i;U_i). \quad (12)$$

The proof of Theorem 2 is presented in Appendix.

Interestingly, this result naturally connects the *online* privacy-utility tradeoff with the *offline* privacy-utility tradeoff, which further enables us to understand the information leakage in the offline setting. In addition, the above result is generic in the sense that $\mathcal{L}_{\text{lower}}(D)$ and $\mathcal{L}_{\text{upper}}(D)$ are true for any kind of online location release mechanism. Moreover, Theorem 2 also provides a framework to pre-compute the optimal LPPM $q(u_i|l_i)$ at every timestamp according to $\mathcal{L}_{\text{upper}}(D)$ and then release location u_i at timestamp i by sampling from $q(u_i|l_i)$. We only need to characterize the values of $q(u_i|l_i)$ for N^2 possible combinations in $\mathcal{L}_{upper}(D)$, which leads to high computation efficiency.

However, we notice that $\mathcal{L}_{\text{upper}}(D)$ only captures the mutual information between the true location L_i and the released location U_i without taking any correlation into account. Therefore, $\mathcal{L}_{\text{upper}}(D)$ may be too weak to be a privacy metric to quantify location trace privacy, where correlations naturally happen.

Last, as mentioned above, Theorem 1 and Corollary 1 are only valid under the Markov release restriction. On the other hand, Theorem 2 provides the generic bounds for any optimal online LPPM, since we did not make the Markov release restriction when deriving the main results in this Theorem.

V. Algorithms for Generating Optimal LPPMs based on Upper Bounds

In this section, we propose algorithms to obtain the optimal LPPMs based on the upper bounds $\mathcal{L}^{\text{Markov}}_{upper}(D)$ (i.e., LPPM*) and $\mathcal{L}_{\text{upper}}(D)$. Since generating LPPM based on $\mathcal{L}_{\text{upper}}(D)$ has already been proposed in [11], we will mainly focus on presenting the algorithm used to solve the optimization problem in $\mathcal{L}^{\text{Markov}}_{upper}(D)$ to generate LPPM*.

Note that the objective function in $\mathcal{L}^{\text{Markov}}_{upper}(D)$ is in the form of conditional mutual information, thus solving the optimization problem in $\mathcal{L}^{\text{Markov}}_{upper}(D)$ comes down to the problem of minimizing conditional mutual information subject to a utility constraint, which can be solved by our proposed algorithm.

A. Algorithm for Minimizing Conditional Mutual Information subject to a Utility Constraint

First, we define the problem of minimizing conditional mutual information subject to a utility constraint as below.

***Definition** 6: If X,$\hat{X}$ and S are random variables, the problem of minimizing conditional mutual information $I(X;\hat{X}|S)$ subject to a utility constraint D is*

$$\min_{q(\hat{x}|x,s):\tilde{D}\leq D} I(X;\hat{X}|S),$$

where $\tilde{D} = \sum_{x,\hat{x},s} p(x|s)q(\hat{x}|x,s)p(s)d(x,\hat{x})$.

Noticing that the problem in Definition 6 is similar to the rate distortion function, therefore, we can follow the main idea of computing the rate distortion function to solve this problem. We start with rewriting the optimization problem in Definition 6 as a minimum of the relative entropy between two sets, and then apply the process of alternating minimization to obtain the optimal solution $q(\hat{x}|x,s)$.

1) Rewrite the Optimization Problem in Definition 6: We begin with the following lemma.

***Lemma** 1: Let $p(x|s)p(\hat{x}|x,s)$ be a given joint distribution. Then the distribution $r(\hat{x}|s)$ that minimizes the relative entropy $D(p(x,s)p(\hat{x}|x,s)||(p(x,s)r(\hat{x}|s))$ is the marginal distribution $r^*(\hat{x}|s)$ corresponding to $p(\hat{x}|x,s)$:*

$$D(p(x,s)p(\hat{x}|x,s)||p(x,s)r^*(\hat{x}|s)) = \min_{r(\hat{x}|s)} D(p(x,s)p(\hat{x}|x,s)||(p(x,s)r(\hat{x}|s)),$$

where $r^*(\hat{x}|s) = \sum_x p(x|s)p(\hat{x}|x,s)$.

The proof of Lemma 1 is similar to the main idea in the proof of Lemma 10.8.1 in [33], thus we omit this proof for space limitation.

According to the definition of mutual information, we have

$$I(X;\hat{X}|S) = \sum_{x,\hat{x},s} p(x,s)q(\hat{x}|x,s)\log\frac{q(\hat{x}|x,s)}{q(\hat{x}|s)}. \quad (13)$$

Based on Lemma 1 and Eq. (13), we can rewrite the optimization problem in Definition 6 as a double minimization as below,

$$\min_{q(\hat{x}|x,s):\tilde{D}\le D} I(X;\hat{X}|S) = \min_{r(\hat{x}|s)} \min_{\substack{q(\hat{x}|x,s):\\ \tilde{D}\le D}} \sum_{x,\hat{x},s} p(x,s)q(\hat{x}|x,s)\log\frac{q(\hat{x}|x,s)}{r(\hat{x}|s)}. \quad (14)$$

2) Alternating Minimization Between Two Sets to Obtain the Optimal Solution: If A is the set of all joint distributions over $X,\hat{X},S$ with marginal $p(x,s)$ that satisfy the distortion constraint and if B is the set of product distributions $p(x,s)r(\hat{x}|s)$ with arbitrary $r(\hat{x}|s)$, we can rewrite Eq. (14) as the following,

$$\min_{q(\hat{x}|x,s):\tilde{D}\le D} I(X;\hat{X}|S) = \min_{q\in B}\min_{p\in A} D(p||q). \quad (15)$$

Until now, we have converted the problem of minimizing conditional mutual information into the problem of finding the minimum of the relative entropy between two sets. Similar to the algorithm for finding the minimum distance between two convex sets mentioned in Section III, we can use an alternating minimization process on the double minimization problem in (15) to obtain the optimal solution $q(\hat{x}|x,s)$.

We begin with an initial output distribution $r(\hat{x}|s)$ and calculate the $q(\hat{x}|x,s)$ that minimizes the conditional mutual information $I(X;\hat{X}|S)$ subject to the distortion constraint. We can use the method of Lagrange multipliers for this minimization to obtain

$$q(\hat{x}|x,s) = \frac{r(\hat{x}|s)e^{-1-\lambda d(x,\hat{x})}}{\sum_{\hat{x}} r(\hat{x}|s)e^{-1-\lambda d(x,\hat{x})}}. \quad (16)$$

The parameter λ in Eq. (16) is the Lagrange multiplier, which is related to how much we favor information leakage versus distortion (higher λ means less distortion).

For this conditional distribution $q(\hat{x}|x,s)$, we calculate the output distribution $r(\hat{x}|s)$ that minimizes the conditional mutual information, which by Lemma 1 is

$$r(\hat{x}|s) = \sum_x p(x|s)q(\hat{x}|x,s). \quad (17)$$

Since the proofs of Eq. (16) and (17) are similar to those of Eq. (10.142) and (10.143) in [33], we omit these proofs for space consideration.

We use the output distribution $r(\hat{x}|s)$ as the starting point of the next iteration. Each step in the iteration, minimizing over $q(\hat{x}|x,s)$ and then over $r(\hat{x}|s)$, reduces the right-hand side of (14). The limit of this process has been shown in [34].

The algorithm used to obtain the optimal solution $q(\hat{x}|x,s)$ to the problem in Definition 6 is presented in Algorithm 1.

Algorithm 1: Minimizing Conditional Mutual Information

Input: λ: Lagrange multiplier, $p(x,s)$: joint probability distribution of x and s, $d(x,\hat{x})$: distortion matrix, δ: threshold for convergence of the algorithm
Output: $q(\hat{x}|x,s)$: optimal solution, I^*: minimum of $I(X;\hat{X}|S)$, D: distortion corresponding to I^*

```
Initialize r_0(x̂|s) as a uniform distribution
Calculate q_0(x̂|x,s) using r_0(x̂|s) by Eq.(16)
Calculate I_0 = I(X;X̂|S) using r_0(x̂|s), q_0(x̂|x,s), and p(x,s) by Eq.(13)
Calculate r(x̂|s) using q_0(x̂|x,s) by Eq.(17)
while true do
    Calculate q(x̂|x,s) using r(x̂|s) by Eq.(16)
    Calculate I = I(X;X̂|S) using r(x̂|s), q(x̂|x,s), and p(x,s) by Eq.(13)
    if (I_0 − I ⩽ δ) then
        I* ← I
        Calculate D = Σ_{x,x̂,s} p(x,s)q(x̂|x,s)d(x,x̂)
        return q(x̂|x,s), I*, D
    else
        I_0 ← I
        Calculate r(x̂|s) using q(x̂|x,s) by Eq.(17)
    end if
end while
```

***Remark** 2: We analyze the computation complexity of Algorithm 1 by giving an expression for the computation complexity of one iteration in the algorithm. In each iteration, the computation complexity is dominated by the calculation for $q(\hat{x}|x,s)$ and $r(\hat{x}|s)$. Complexity of computing $q(\hat{x}|x,s)$ as given in (16): For each combination of (x,s), we need $|\hat{X}|$ multiplications for a specific $|\hat{x}|$ in the denominator and then use this denominator for every other $|\hat{x}|$, thus the calculation needs $O(|\hat{X}|)$ operations. Considering of all the combinations of (x,s), the complexity of computing $q(\hat{x}|x,s)$ will be $O(|\hat{X}||X||S|)$. Complexity of computing $r(\hat{x}|s)$ as given in (17): Similarly, for each s, we need $|X|$ multiplications for a specific $|\hat{x}|$, so the calculation for all the s requires $O(|X||S|)$ operations for a specific $|\hat{x}|$. Considering of the calculation for all the $|\hat{x}|$, the complexity of computing $r(\hat{x}|s)$ will also be $O(|\hat{X}||X||S|)$. Therefore, each iteration in Algorithm 1 requires about $O(|\hat{X}||X||S|)$ computations. We can easily see that the computation complexity grows with the size of $\hat{X}$,X and S. However, in practice, it is almost impossible for a user to move to a location far away from her current location within a certain time period due to the speed constraint. As a*

result, when we use Algorithm 1 to generate LPPMs, the size of $\hat{X}$,X and S will not be arbitrarily large and are actually limited to a small region.

In the following, we show how to leverage Algorithm 1 as a building block to generate LPPM*, i.e., by generating LPPM$_1^*$,...,LPPM$_i^*$,...,LPPM$_T^*$ in an online manner.

B. Generating LPPMs based on the Upper Bounds

1) Generating LPPM$_i^*$: Now we leverage algorithm 1 to design the algorithm used to generate LPPM$_i^*$ $q(u_i|l_i, u_{i-1}, l_{i-1})$. Basically, we replace X, $\hat{X}$ and S with (L_i, L_{i-1}), U_i and U_{i-1} respectively in Eq.(16) and Eq.(17) to obtain the $q(\hat{x}|x, s)$ and $r(\hat{x}|s)$ used in Algorithm 1, which are shown as the following:

$$q(u_i|l_i, u_{i-1}, l_{i-1}) = \frac{r(u_i|u_{i-1})e^{-1-\lambda d(l_i,u_i)}}{\sum_{u_i} r(u_i|u_{i-1})e^{-1-\lambda d(l_i,u_i)}}, \quad (18)$$

$$r(u_i|u_{i-1}) = \sum_{l_i,l_{i-1}} p(l_i, l_{i-1}|u_{i-1})q(u_i|l_i, u_{i-1}, l_{i-1}). \quad (19)$$

From Eq.(19), we can see that the essential step for calculating $r(u_i|u_{i-1})$ is to calculate $p(l_i, l_{i-1}|u_{i-1})$, which equals to $p(u_{i-1}, l_{i-1}, l_i)$ / $p(u_{i-1})$, where $p(u_{i-1})$ is the marginal distribution of the released locations at timestamp $i-1$. Before we present the algorithm for generating the optimal $q(u_i|l_i, u_{i-1}, l_{i-1})$, we want to show how to calculate $p(u_{i-1}, l_{i-1}, l_i)$ first, since this is an essential step to understand what should be the input parameters of our algorithm. More importantly, this calculation also indicates that our optimal LPPM for a location trace (i.e., LPPM*) should be generated sequentially in terms of time span, i.e., the optimal LPPM at timestamp i (i.e., LPPM$_i^*$ $q(u_i|l_i, u_{i-1}, l_{i-1})$) depends on the optimal LPPM at the previous timestamp (i.e., LPPM$_{i-1}^*$ $q(u_{i-1}|l_{i-1}, u_{i-2}, l_{i-2})$), which has already been highlighted in Section IV-B. For example, if we want to generate the optimal LPPM for a location trace with length 4, first we should generate LPPM$_1^*$ $q(u_1|l_1)$ at the first timestamp, then we can generate LPPM$_2^*$ $q(u_2|l_2, u_1, l_1)$ based on LPPM$_1^*$ at the second timestamp, and the processes at timestamps 3 and 4 are similar to this. Now, we expand the calculation for $p(u_{i-1}, l_{i-1}, l_i)$ as the following:

$$p(u_{i-1}, l_{i-1}, l_i) = p(u_{i-1}, l_{i-1})p(l_i|u_{i-1}, l_{i-1}) \quad (20)$$

$$= p(u_{i-1}, l_{i-1})p(l_i|l_{i-1}) \quad (21)$$

$$= \sum_{u_{i-2},l_{i-2}} p(u_{i-1}, l_{i-1}, u_{i-2}, l_{i-2})p(l_i|l_{i-1}) \quad (22)$$

In the probability calculation above, the reason that (20) equals to (21) is because of the property of the first order markov transition in location traces, i.e., the current location l_i only depends on the previous location l_{i-1}. Now we can easily see that $p(u_{i-1}, l_{i-1}, l_i)$ can be directly calculated by the joint probability distribution in the previous timestamp $p(u_{i-1}, l_{i-1}, u_{i-2}, l_{i-2})$ (i.e., $q(u_{i-1}|l_{i-1}, u_{i-2}, l_{i-2})p(l_{i-1}, u_{i-2}, l_{i-2})$) and the location transition probability $p(l_i|l_{i-1})$. Therefore, in order to generate LPPM$_i^*$ at the current timestamp i, we need to use one of the outputs (e.g., $p(u_{i-1}, l_{i-1}, u_{i-2}, l_{i-2})$) from the previous timestamp $i-1$, i.e., this proves the statement above that the optimal LPPM for an entire location trace (i.e., LPPM*) should be generated sequentially in terms of time span.

Algorithm 2: Generating LPPM$_i^*$ at timestamp i

Input: λ: Lagrange multiplier, $p(u_{i-1}, l_{i-1}, u_{i-2}, l_{i-2})$: joint probability distribution obtained from the previous timestamp $i-1$, $p(u_{i-1})$: marginal distribution of the released location u_{i-1}, $d(l_i, u_i)$: distortion matrix, δ: threshold for convergence of the algorithm

Output: $q(u_i|l_i, u_{i-1}, l_{i-1})$: LPPM$_i^*$ at timestamp i, I_i^*: minimum leakage at timestamp i, D_i: distortion corresponding to I_i^*, $p(u_i)$: marginal distribution of the released location u_i, $q(u_i, l_i, u_{i-1}, l_{i-1})$: joint probability distribution of the true and released locations $u_i, l_i, u_{i-1}, l_{i-1}$

Initialize $r_0(u_i|u_{i-1})$ as a uniform distribution
Calculate $q_0(u_i|l_i, u_{i-1}, l_{i-1})$ using $r_0(u_i|u_{i-1})$ by Eq.(18)
Calculate $p(l_i, u_{i-1}, l_{i-1})$ by Eq.(22)
Calculate $I_i^0 = I(L_i, L_{i-1}; U_i|U_{i-1})$ using $r_0(u_i|u_{i-1})$, $q_0(u_i|l_i, u_{i-1}, l_{i-1})$, and $p(l_i, u_{i-1}, l_{i-1})$ by Eq.(23)
Calculate $r(u_i|u_{i-1})$ using $q_0(u_i|l_i, u_{i-1}, l_{i-1})$, $p(l_i, u_{i-1}, l_{i-1})$ and $p(u_{i-1})$ by Eq.(19)
while true **do**
  Calculate $q(u_i|l_i, u_{i-1}, l_{i-1})$ using $r(u_i|u_{i-1})$ by Eq.(18)
  Calculate $I_i = I(L_i, L_{i-1}; U_i|U_{i-1})$ using $r(u_i|u_{i-1})$, $q(u_i|l_i, u_{i-1}, l_{i-1})$, and $p(l_i, u_{i-1}, l_{i-1})$ by Eq.(23)
  if $(I_i^0 - I_i \leqslant \delta)$ **then**
    $I_i^* \leftarrow I_i$
    Calculate $p(u_i, l_i, u_{i-1}, l_{i-1}) = q(u_i|l_i, u_{i-1}, l_{i-1})p(l_i, u_{i-1}, l_{i-1})$
    Calculate $D_i = \sum_{u_i,l_i,u_{i-1},l_{i-1}} p(u_i, l_i, u_{i-1}, l_{i-1})d(l_i, u_i)$
    Calculate $p(u_i) = \sum_{l_i,u_{i-1},l_{i-1}} p(u_i, l_i, u_{i-1}, l_{i-1})$
    return $q(u_i|l_i, u_{i-1}, l_{i-1})$, I_i^*, D_i, $p(u_i, l_i, u_{i-1}, l_{i-1})$ and $p(u_i)$
  else
    $I_i^0 \leftarrow I_i$
    Calculate $r(u_i|u_{i-1})$ using $q(u_i|l_i, u_{i-1}, l_{i-1})$, $p(l_i, u_{i-1}, l_{i-1})$ and $p(u_{i-1})$ by Eq.(19)
  end if
end while

After we obtain LPPM$_i^*$ at each timestamp i, we can calculate its minimum information leakage as below,

$$I(L_i, L_{i-1}; U_i|U_{i-1}) = \sum_{\substack{u_i,l_i,\\ u_{i-1},l_{i-1}}} p(u_i, l_i, u_{i-1}, l_{i-1}) \log \frac{q(u_i|l_i, u_{i-1}, l_{i-1})}{r(u_i|u_{i-1})}. \quad (23)$$

We present the algorithm used to generate LPPM$_i^*$ at timestamp i in a location trace in Algorithm 2.

By adding up all the I_i^* and D_i derived at each timestamp i, we obtain the $\mathcal{L}_{upper}^{\text{Markov}}(D)$ (i.e., an information leakage-distortion pair in the form of $(\sum_{i=1}^T I_i^*, \sum_{i=1}^T D_i)$) of an entire

trace for a fixed λ. By changing to different λs, we can obtain the privacy-utility tradeoff curve of $\mathcal{L}_{upper}^{\text{Markov}}(D)$.

Last, the computation complexity of Algorithm 2 can be analyzed similarly as in Remark 2. Therefore, each iteration in Algorithm 2 requires about $O(|U_i||L_i||U_{i-1}||L_{i-1}|)$ computations.

2) Generating Optimal LPPM based on $\mathcal{L}_{upper}(D)$: As mentioned before, generating the optimal LPPM $q(u_i|l_i)$ based on $\mathcal{L}_{\text{upper}}(D)$ which only protects the current location, i.e., deriving the optimal solution by solving the ith optimization problem in the summation in (11), has already been done in [11]. Therefore, details about the generating process of this LPPM can be referred to [11].

Since the LPPM generated based on $\mathcal{L}_{\text{upper}}(D)$ does not capture location correlations, this may not be very helpful when designing LPPM for location traces. However, this method works very well in the single location scenario (i.e., the released location u_i only depends on the current true location l_i at timestamp i), since it can guarantee the minimum information leakage. Besides, the authors in [11] have already shown its advantage over other schemes. Later, instead of evaluating the optimal LPPM $q(u_i|l_i)$ generated based on $\mathcal{L}_{\text{upper}}(D)$, we will show the results for the LPPM proposed in [11] in Section VI, because they are the same.

***Remark** 3: Since the lower bounds $\mathcal{L}_{lower}^{Markov}(D)$ and $\mathcal{L}_{lower}(D)$ only capture the least amount of information leakage when user releases her location trace, the actual leakage may be higher than the leakage quantified by $\mathcal{L}_{lower}^{Markov}(D)$ and $\mathcal{L}_{lower}(D)$. Therefore, generating LPPMs based on $\mathcal{L}_{lower}^{Markov}(D)$ and $\mathcal{L}_{lower}(D)$ may not be meaningful in the sense that they cannot provide any privacy guarantee about the upper bound of the information leakage. However, for completeness, we still calculate the $\mathcal{L}_{lower}^{Markov}(D)$ in Section VI to help us have a clear understanding about the actual information leakage when user releases her distorted location trace. Technically, $\mathcal{L}_{lower}^{Markov}(D)$ can be solved based on Algorithm 1. For space consideration, we will not show the results for $\mathcal{L}_{lower}(D)$.*

C. Evaluating LPPMs based on the Actual Online Leakage

We use the actual online leakage $L_{\text{online}}^{\text{Actual}}(\text{LPPM})$ in definition 4 as the privacy metric to evaluate our LPPM and other proposed LPPMs. This actual online leakage can be used as a general metric to evaluate existing and future LPPMs. Now we describe how to use this metric to evaluate LPPMs by expanding the formulation of $L_{\text{online}}^{\text{Actual}}(\text{LPPM})$ as the following,

$$\begin{aligned} &L_{\text{online}}^{\text{Actual}}(\text{LPPM}) = \sum_{i=1}^{T} I(L_1,...,L_i;U_i|U_1,...,U_{i-1}) \\ &=\sum_{i=1}^{T}(H(U_i|U_1,...,U_{i-1}) - H(U_i|U_1,...,U_{i-1},L_1,...,L_i)). \end{aligned} \tag{24}$$

We want to highlight that the calculation for both terms in (24) can be simplified depending on how a specific LPPM is designed. We take the calculation for $L_{\text{online}}^{\text{Actual}}(\text{LPPM})$ of our proposed LPPM (i.e., LPPM*) and the LPPM proposed in [11] as examples to illustrate this point.

For the second term in (24), because we have made the restriction on location releasing mechanism that U_i only depends on L_i, U_{i-1} and L_{i-1}, we have $H(U_i|U_1,...,U_{i-1},L_1,...,L_i) = H(U_i|L_i,U_{i-1},L_{i-1})$ in (i.e., LPPM$_i^*$); since the LPPM proposed in [11] is used in the single location scenario, then we have $H(U_i|U_1,...,U_{i-1},L_1,...,L_i) = H(U_i|L_i)$ in their scheme. Similar simplification can also be made when evaluating other existing or future LPPMs which are designed in different ways. We know that calculating $H(U_i|L_i,U_{i-1},L_{i-1})$ only requires the conditional probability distribution $q(u_i|l_i,u_{i-1},l_{i-1})$ (i.e., LPPM$_i^*$) and the joint distribution $p(u_i,l_i,u_{i-1},l_{i-1})$, which can be obtained from Algorithm 1. Similarly, the calculation for $H(U_i|L_i)$ only depends on conditional probability distribution $q(u_i|l_i)$ (i.e., the optimal LPPM in [11]) and the joint distribution $p(u_i,l_i)$.

As for the first term in (24), the computation complexity for its calculation grows exponentially with the location trace length T, since this calculation needs the joint distribution for all the released locations $u_1,u_2...,u_i$. As an illustrating example, we consider calculating the actual online leakage for a location trace with length 3 (i.e., $T=3$). As for a longer location trace, the calculation needs more time.

When $T=3$, we have $\sum_{i=1}^{3} H(U_i|U_1,...,U_{i-1}) = H(U_1) + H(U_2|U_1) + H(U_3|U_2,U_1)$. We know that $H(U_1) = -\sum_{u_1} p(u_1)\log p(u_1)$, $H(U_2|U_1) = -\sum_{u_1,u_2} p(u_2|u_1)p(u_1)\log p(u_2|u_1)$, and $H(U_3|U_2,U_1) = -\sum_{u_1,u_2,u_3} p(u_3|u_1,u_2)p(u_2|u_1)p(u_1)\log p(u_3|u_1,u_2)$. We can see that the essential parts for those calculation are calculating $p(u_2|u_1)$ and $p(u_3|u_1,u_2)$.

For our scheme, we have

$$\begin{aligned} p(u_2|u_1) &= \sum_{l_1,l_2} p(u_2|u_1,l_1,l_2)p(l_1,l_2|u_1) \\ &= \sum_{l_1,l_2} \frac{p(u_2|u_1,l_1,l_2)p(u_1|l_1)p(l_1,l_2)}{p(u_1)}, \end{aligned} \tag{25}$$

and similarly we can also obtain

$$\begin{aligned} &p(u_3|u_1,u_2) \\ &= \sum_{l_1,l_2,l_3} \frac{p(u_3|u_2,l_2,l_3)p(u_2|u_1,l_1,l_2)p(u_1|l_1)p(l_1,l_2,l_3)}{p(u_2|u_1)p(u_1)}. \end{aligned} \tag{26}$$

From (25) and (26), we can see that the calculation only needs all the LPPMs derived up to the current timestamp i, i.e., $p(u_3|u_2,l_2,l_3)$, $p(u_2|u_1,l_1,l_2)$ and $p(u_1|l_1)$, and the joint probability distribution of all the true locations $p(l_1,l_2,l_3)$.

Similarly, for the scheme in [11], the calculations are

$$p(u_2|u_1) = \sum_{l_1,l_2} \frac{p(u_2|l_2)p(u_1|l_1)p(l_1,l_2)}{p(u_1)}, \tag{27}$$

$$p(u_3|u_1,u_2) = \sum_{l_1,l_2,l_3} \frac{p(u_3|l_3)p(u_2|l_2)p(u_1|l_1)p(l_1,l_2,l_3)}{p(u_2|u_1)p(u_1)}. \tag{28}$$

The calculation in (27) and (28) also only requires all the LPPMs derived up to the current timestamp i, i.e., $p(u_3|l_3)$,

$p(u_2|l_2)$ and $p(u_1|l_1)$, and the joint probability distribution of all the true locations $p(l_1, l_2, l_3)$.

Finally, by combining the calculation for both terms in (24), we can calculate the actual online leakage for any type of LPPMs. Therefore, the actual online leakage $L^{\text{Actual}}_{\text{online}}(\text{LPPM})$ can be used as a generic privacy metric to evaluate and compare existing and future LPPMs, which is extremely meaningful in terms of designing privacy-preserving mechanisms. The calculation of I_{upper} and I_{lower} can be done similarly as the process presented above.

VI. Experimental Evaluation

In this section, we evaluate the actual online leakage $L^{\text{Actual}}_{\text{online}}(\text{LPPM})$ of different LPPMs, $L^{\text{Markov}}_{upper}(D)$ and $L^{\text{Markov}}_{\text{lower}}(D)$ on a synthetic dataset; also evaluate $L^{\text{Markov}}_{upper}(D)$ and $L^{\text{Markov}}_{\text{lower}}(D)$ on a real-world dataset. Specifically, we evaluate the actual online leakage of our LPPM (i.e., LPPM*), the LPPM proposed in [11] (denoted by OTP17) and another LPPM proposed in [10] (denoted by ABCP13). OTP17 has been briefly mentioned in Section V-B2, and ABCP13 is based on differential privacy and also works in single location scenario. We did not evaluate the LPPM proposed in [17] which considers temporal correlations when protecting location privacy, since deriving the probability distributions for calculating the actual online leakage from their LPPM is non-trival. All experiments were conducted on a desktop with 3.6 GHz Intel i7 CPU and 8GB memory.

A. Evaluation on Synthetic Dataset

Since our privacy metric defined as the actual online leakage $L^{\text{Actual}}_{\text{online}}(\text{LPPM})$ is designed for evaluating trace-level privacy, intuitively, this metric should capture different correlation levels among location traces inherently. In order to explore how different correlation levels will affect the privacy leakage, we need to make sure that the location traces have the intended correlation levels by design. Therefore, we start with a synthetic dataset to properly evaluate this effect. We use specifically designed Markov models on a synthetic dataset to generate location traces. An illustration of the synthetic dataset is presented in Fig. 3. In this dataset, we consider a map with 6 locations, which is divided by 2×3 grids, and the width and length of each grid are both defined as 1 without loss of generality. The number located inside a grid is an index for that grid/location; for example, the index of the location $(3, 2)$ is 6.

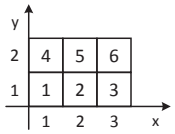


Fig. 3. Synthetic dataset: a map with 6 locations.

In this synthetic dataset, we use four different types of Markov transition matrices to model the different correlation levels among location traces. These are all 6×6 matrices and denoted by $\boldsymbol{M}_1$, $\boldsymbol{M}_2$, $\boldsymbol{M}_3$ and $\boldsymbol{M}_4$ respectively. Specifically, each row in $\boldsymbol{M}_1$ has one element as 1 and the others as 0; each row in $\boldsymbol{M}_2$ has two elements as $1/2$ and the others as 0; all elements in $\boldsymbol{M}_3$ are generated randomly and then each row is normalized to form a probability distribution; all elements in $\boldsymbol{M}_4$ are the same (i.e., $1/6$). Then $\boldsymbol{M}_1$, $\boldsymbol{M}_2$, $\boldsymbol{M}_3$ and $\boldsymbol{M}_4$ can be used to generated location traces which are correlated in a decreased manner, i.e., location traces generated based on $\boldsymbol{M}_1$ are fully correlated, while traces generated based on $\boldsymbol{M}_4$ are fully independent.

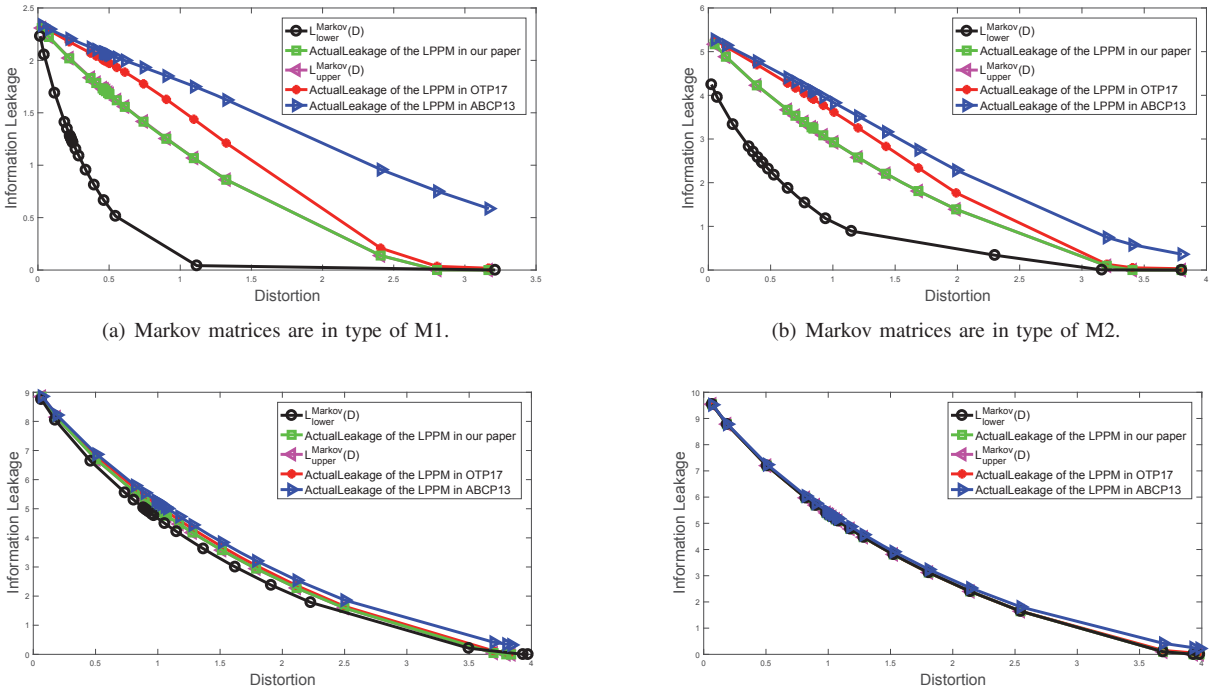
Besides, we adopt Euclidean distance as the distortion function since the distortion between two locations is sensitive to the their distance, i.e., $d(l, u) = \sqrt{|(l_x - u_x)^2 + (l_y - u_y)^2|}$, where (l_x, l_y) and (u_x, u_y) are the two coordinates for location l and u respectively.

We consider a location trace with length of 4, i.e., $\boldsymbol{L} = \{L_1, L_2, L_3, L_4\}$. Each location is represented by an index k in $\{1, 2, ..., 6\}$. Given the location distribution for the 1st timestamp p_1 as $\{p_1^1, p_1^2, p_1^3, p_1^4, p_1^5, p_1^6\}$ (p_i^k represents the probability when user's location index is k at the ith timestamp), the location probability distribution for the subsequent timestamps can be calculated based on p_1 and the Markov transition matrix $\boldsymbol{M}$ by equation $p_i = p_{i-1}\boldsymbol{M}$. We generate location traces for the synthetic dataset according to the joint probability distribution of all the locations, i.e., $p(l_1, l_2, l_3, l_4)$.

Based on this synthetic dataset, we show the performance of $\mathcal{L}^{\text{Markov}}_{upper}(D)$ and $\mathcal{L}^{\text{Markov}}_{\text{lower}}(D)$, and evaluate the actual online leakage of the LPPM proposed in our paper (i.e., LPPM*), the LPPMs in OTP17 and ABCP13. In particular, the later two LPPMs are derived under the same initial location probability distribution, Markov mobility model and distortion as ours. Besides, we choose λ from the range of 0.01 to 10 to draw the privacy-utility tradeoff curve. Since we know that the less λ is, the larger the distortion. Therefore, by choosing λ in an increasing manner from the range, we can draw privacy-utility tradeoff curve quite smoothly (one point in the figures corresponds to one λ). We set the threshold for convergence in Algorithm 2 as 10^{-8}. All experiments are conducted average on 5 Markov matrices, and the results are shown in Fig. 4.

Now we describe how we derive the curves shown in the figures in detail. Remember that each point on the curve is corresponding to a fixed λ. For a fixed λ, given the initial location probability distribution and a Markov transition matrix, we can obtain the information leakage-distortion pair (i.e., $(\sum_{i=1}^{4} I_i^*, \sum_{i=1}^{4} D_i)$) of $\mathcal{L}^{\text{Markov}}_{upper}(D)$ by Algorithm 2. For each λ, we save the distortion D_i which is one of the outputs of Algorithm 2 at each timestamp i. When changing to different λs, we can smoothly draw the information leakage-distortion curve of $\mathcal{L}^{\text{Markov}}_{upper}(D)$. For every λ, we use D_i, the same initial location probability distribution and Markov transition matrix as inputs to derive the LPPMs in OTP17 and ABCP13 at each timestamp, and then calculate their actual online leakage for the entire trace following the calculation procedure presented in Section V-C. By enumerating all the λs, we derive the actual online leakage curves for the LPPMs in OTP17 and ABCP13. In addition, we also calculate the actual online leakage of the LPPM proposed in our paper (i.e., LPPM*) and compare it with the actual leakage of the other two LPPMs. Similarly, we can also derive the curve of $\mathcal{L}^{\text{Markov}}_{\text{lower}}(D)$.

As we can see from Fig. 4, the actual leakage of the LPPM proposed in our paper (i.e., LPPM*) is lower than the other two LPPMs in all the cases we considered. In particular, this



(a) Markov matrices are in type of M1.

(b) Markov matrices are in type of M2.

(c) Markov matrices are in type of M3.

(d) Markov matrices are in type of M4.

Fig. 4. $\mathcal{L}^{\text{Markov}}_{upper}(D)$, $\mathcal{L}^{\text{Markov}}_{\text{lower}}(D)$, and actual leakage of different LPPMs evaluated under different types of Markov matrices.

advantage becomes even greater when location trace is more correlated. The reason that LPPM* has the least information leakage is because it considers U_{i-1} when releasing locations. Specifically, when the comparison was done on $M1$, the location trace is mostly a single static location over time (i.e., $L_i = L_{i-1}$), if we do not consider U_{i-1}, the LPPM will output a new U_i as a noisy version of L_i (which equals to L_{i-1}) independent of the previous U_{i-1}, with the same probability distribution as U_{i-1} since the inputs are the same, so an attacker can better infer L_i by combining the two noisy outputs. In this case, our leakage will be similar to the LPPM in OTP17 since they only consider L_i in their LPPM. But if we consider U_{i-1}, we can now output the same value $U_i = U_{i-1}$, which will still satisfy the distortion constraint but less leakage than the case which do not consider U_{i-1}. Since the the LPPM in ABCP13 considers neither correlations nor minimizing information leakage, its leakage is the largest compared with our LPPM and the one in OTP17.

Besides, the actual leakage-distortion curve of our LPPM is always upper bounded by $\mathcal{L}^{\text{Markov}}_{\text{upper}}(D)$, which follows the main result from Corollary 1. In other words, the experimental results further strengthen the statement that our LPPM can provide the privacy guarantee that its actual leakage is sandwiched between $\mathcal{L}^{\text{Markov}}_{\text{lower}}(D)$ and $\mathcal{L}^{\text{Markov}}_{\text{upper}}(D)$. Therefore, even though generating LPPM for location traces directly from $\mathcal{L}^{*}_{\text{online}}(D)$ is exponentially expensive, we can still use $\mathcal{L}^{\text{Markov}}_{\text{upper}}(D)$ to efficiently generate LPPM* with strict privacy guarantee and remain lower leakage compared with other LPPMs in terms of trace-level privacy.

We want to highlight one important take away from the results in Fig. 4, which is that the actual leakage is very close to the upper bound $\mathcal{L}^{\text{Markov}}_{\text{upper}}(D)$. This is intuitive since $\mathcal{L}^{\text{Markov}}_{\text{upper}}(D)$ generates the LPPM in a way such that the released location U_i at timestamp i depends on the current true location L_i, the previous released location U_{i-1}, and the previous true location L_{i-1}; and we only consider the first-order Markov mobility model (i.e., the current location L_i only depends on the previous location L_{i-1}), therefore, designing LPPM based on $\mathcal{L}^{\text{Markov}}_{\text{upper}}(D)$ may be already enough to capture the full correlations among locations, which leads to the actual leakage of our LPPM to be quite close to the upper bound $\mathcal{L}^{\text{Markov}}_{\text{upper}}(D)$.

In addition, to understand how LPPM* works in an intuitive way, we analyze how the location traces are distorted based on LPPM*. Specifically, we show the true location trace and its corresponding distorted location trace (both of length of 8) on $\boldsymbol{M}_1$ and $\boldsymbol{M}_4$ when $\lambda = 0.5$ in Fig. 5. We can see that when the Markov transition matrix is $\boldsymbol{M}_1$, the distorted locations remain almost the same with the true locations. Remember that the user's locations are highly correlated when the Markov transition matrix is $\boldsymbol{M}_1$. In this case, it is not necessary to distort the locations by a large extent, because this will decrease the utility and is not very helpful to decrease the information leakage. The reason for not being helpful to decrease the information leakage is that the initial location distribution and the Markov transition matrix have already leak a significant amount of information, even though we do not release the distorted locations. Hence, LPPM* ensures good utility by not distorting the locations by a large amount in this case. When the Markov transition matrix is $\boldsymbol{M}_4$, the true locations are distorted by a very large extent to ensure less information leakage, only very few distorted location remains the same with the true location. The results are intuitive since the optimization problem used to generate LPPMs in $L^{\text{Markov}}_{upper}(D)$ consider information leakage and utility at the same time.

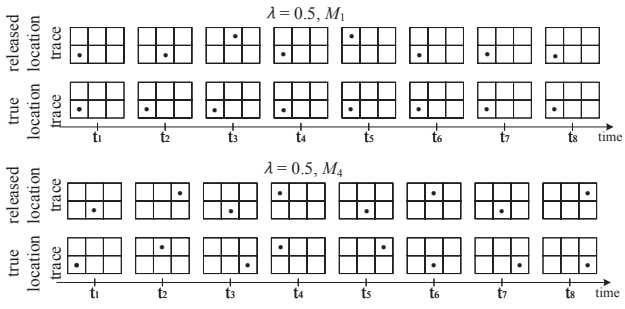


Fig. 5. True location trace and released location trace generated from our LPPM on $\boldsymbol{M}_1$ and $\boldsymbol{M}_4$ when λ = 0.5.

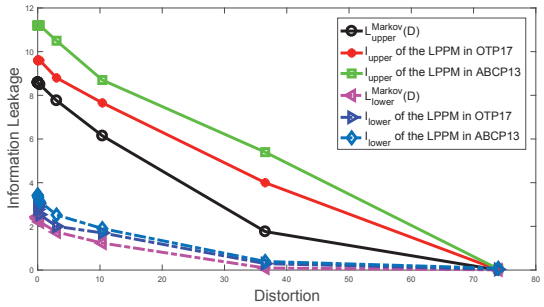


Fig. 6. $L^{\text{Markov}}_{\text{upper}}(D)$ and $L^{\text{Markov}}_{\text{lower}}(D)$ of our LPPM, I_{upper} and I_{lower} of other LPPMs on the real-world dataset.

B. Evaluation on Real-world Dataset

Since directly calculating the actual online leakage $L^{\text{Actual}}_{\text{online}}(\text{LPPM})$ on a real-world dataset leads to exponential complexity, in this subsection, we evaluate $L^{\text{Markov}}_{upper}(D)$ and $L^{\text{Markov}}_{\text{lower}}(D)$ based on a real-world dataset Geolife [35] collected by Microsoft Research Asia. Geolife dataset contains 182 users' location trace data in a period of over five years. In this dataset, user's GPS trace is represented by a sequence of tuples, each of which contains latitude, longitude and timestamp. The majority of this dataset was created in Beijing, China. In our experiment, we pre-processed the dataset by extracting a part of the data within the 3rd ring of Beijing of size $7.2km \times 7.2km$, and divided this area into 12×12 grids of $0.6km \times 0.6km$. For simplicity, we evaluate user's location trace with length 8. Markov transition matrix for this real-world dataset is learned by the EM method [36]. We also adopt Euclidean distance to calculate the distortion function. We randomly choose 5 users out of the entire dataset to train their personal Markov transition matrices and location probability distributions, and the results are averaged on those 5 users. We choose λ from the same range with the one used in the synthetic dataset and set the threshold as 0.0001.

1) Evaluating $L^{Markov}_{upper}(D)$, $L^{Markov}_{lower}(D)$, and I_{upper} and I_{lower} of other LPPMs: After we derive LPPM* on the real-world dataset, by using similar process as the experiments in the synthetic dataset, we can use the same distortion to derive the LPPMs in OTP17 and ABCP13. Since we know that calculating the actual online leakage on the real-world dataset leads to exponential complexity even for the LPPMs proposed for protecting single location, we calculate I_{upper} and I_{lower} for the LPPMs in OTP17 and ABCP13. We show the results in Fig. 6. From this figure, the first thing we know is that $L^{\text{Markov}}_{upper}(D)$ is lower than the I_{upper} of the other two LPPMs, and $L^{\text{Markov}}_{\text{lower}}(D)$ is lower than the I_{lower} of the other two LPPMs. Even though we cannot directly compare the actual online leakage of these LPPMs because of the exponential complexity problem, we still know that the actual online leakage of LPPM* should be very close to $L^{\text{Markov}}_{upper}(D)$, based on the results shown in synthetic dataset. Besides, we know that the information leakage of LPPM* should be sandwiched between $L^{\text{Markov}}_{upper}(D)$ and $L^{\text{Markov}}_{\text{lower}}(D)$, since we have theoretically proved that the actual online leakage is sandwiched between the upper and lower bounds in Corollary 1. In addition, the actual online leakage of LPPM* should be no worse than the one in OTP17 from a theoretical point of view, this is because we are trying to minimize privacy leakage over a larger distribution $q(u_i|l_i, u_{i-1}, l_{i-1})$ instead of a smaller distribution $q(u_i|l_i)$. Last, we want to highlight that even though we cannot evaluate the actual online leakage of LPPM* on the real-world dataset directly due to the exponential complexity problem, our advantage is that LPPM* can still provide reasonable privacy guarantee in a scenario where locations are correlated, since its actual online leakage is no larger than $L^{\text{Markov}}_{upper}(D)$.

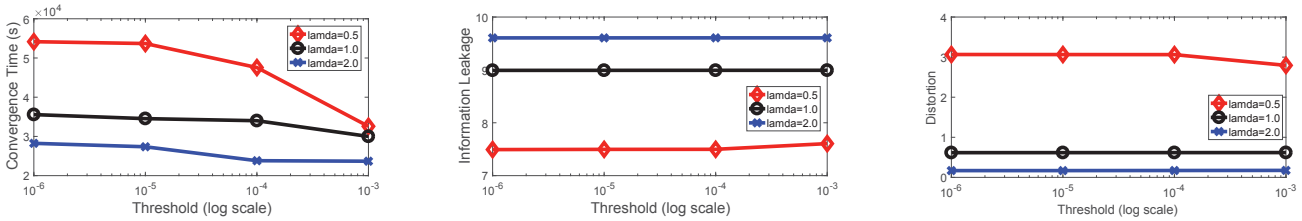
2) Impact of Threshold on Real-world Dataset: We also analyze the computation time for Algorithm 2 on real-world dataset, i.e., the time that Algorithm 2 needs to converge. As we can see from Algorithm 2, when fixing the trace length and the size of locations, the time of convergence only depends on the threshold. To evaluate how the time of convergence changes with different thresholds, we run our algorithm on 4 different thresholds, i.e., 0.000001, 0.00001, 0.0001 and 0.001 respectively when $\lambda = 0.5, 1.0, 2.0$. Besides, when Algorithm 2 converges at different iterations, the minimum information leakage and the corresponding distortion may also be different and thus need to be evaluated to see the extent of differences.

As we can see from Fig. 7, with the increase of threshold from 0.000001 to 0.001, the time of convergence for Algorithm 2 decreases by a large extent when $\lambda = 0.5, 1.0, 2.0$, however, the minimum information leakage and the corresponding distortion change very slightly. This means that a user can choose a slightly larger threshold to reduce a large amount of computation time while only sacrificing little privacy.

In addition, we can also know the approximate time for pre-computing LPPM* to release a location trace with length of 8 from Fig. 7(a), which is around $22,000$ seconds (i.e., about 6 hours, i.e., 45 minutes for each location on average). We argue that this pre-computation time is reasonable, since a user usually releases her location trace at daytime and does not do so during nighttime, thus the pre-computation can be finished at night based on her current initial location distribution and Markov transition matrix. As a user does not change her mobility profile very often, once the pre-computation is completed, the user can release her distorted location trace using the pre-computed LPPM* very efficiently, until her mobility profile is changed.

VII. Conclusion and Future Work

We have proposed privacy metrics to quantify location trace privacy independent of any specific attack based on an information-theoretic approach in both an offline setting and



(a) Impact of threshold on convergence time. (b) Impact of threshold on information leakage. (c) Impact of threshold on distortion.

Fig. 7. Impact of threshold.

an online setting. Specifically, our privacy metrics quantify the inherent information leakage when releasing user's distorted location trace instead of her true location trace to the untrusted service provider. We have formulated the problem to obtain the optimal LPPM used to release user's locations to get services while achieving the minimum information leakage in an online setting. To address the computation challenge occurred when computing the optimal LPPM directly from the online problem, we obtain the upper and lower bounds on the online privacy-utility tradeoff with and without Markov restriction on release mechanisms, and thus we can obtain the LPPM based on the upper bounds with very high efficiency. In particular, the offline privacy-utility tradeoff has a natural connection with the online tradeoff when there is no Markov release restriction. Experiments have shown the advantage of our LLPM over existing LLPMs in terms of privacy-utility tradeoff, which is greater when the location trace is more correlated. In addition, the proposed privacy metrics can also be used as standard measures to evaluate and compare other privacy-preserving mechanisms for time-series data and not limited to LBS, which is very meaningful in many real-world applications. We direct this to future work.

APPENDIX

A. Proof of $\mathcal{L}^*_{offline}(D) \leq \mathcal{L}^*_{online}(D)$

We start with proving the connection between the objective function of $\mathcal{L}^*_{\text{offline}}(D)$ and the summation of the objective functions in $\mathcal{L}^*_{\text{online}}(D)$ as follows,

$$\begin{aligned}
&I(\boldsymbol{L};\boldsymbol{U})\\
\overset{(a)}{=}&\sum_{i=1}^{T} I(\boldsymbol{L};U_i|\boldsymbol{U}^{i-1})\\
=&\sum_{i=1}^{T}\{I(\boldsymbol{L}^i;U_i|\boldsymbol{U}^{i-1}) + I(L_{i+1},...,L_T;U_i|\boldsymbol{U}^{i-1},\boldsymbol{L}^i)\}\\
\overset{(b)}{=}&\sum_{i=1}^{T} I(\boldsymbol{L}^i;U_i|\boldsymbol{U}^{i-1}),
\end{aligned}$$

where (a) follows from the chain rule of mutual information, and (b) follows from the fact in an online location release setting, which is the current released location U_i is independent of the future true locations $L_{i+1},...,L_T$ given $\boldsymbol{U}^{i-1},\boldsymbol{L}^i$.

As we can see from the above proof, the objective function in $\mathcal{L}^*_{\text{offline}}(D)$ is equal to the summation of the objective functions in $\mathcal{L}^*_{\text{online}}(D)$. Since the variables where the optimization takes place are different for each term of the summation, we can conclude that minimizing the summation is less than or equal to the summation of the individual minimizations, thus we have $\mathcal{L}^*_{\text{offline}}(D) \leq \mathcal{L}^*_{\text{online}}(D)$.

B. Proof of Theorem 1

1) $\mathcal{L}^*_{online}(D) \leq \mathcal{L}^{Markov}_{upper}(D)$: We prove the upper bound of the objective function $I(\boldsymbol{L}^i;U_i|\boldsymbol{U}^{i-1})$ as follows,

$$\begin{aligned}
&I(\boldsymbol{L}^i;U_i|\boldsymbol{U}^{i-1})\\
\overset{(c)}{=}&I(L_{i-1},L_i;U_i|U_1,U_2,...,U_{i-1})\\
&+I(L_1,L_2,...,L_{i-2};U_i|U_1,U_2,...,U_{i-1},L_{i-1},L_i)\\
\overset{(d)}{=}&I(L_{i-1},L_i;U_i|U_1,U_2,...,U_{i-1})\\
=&H(U_i|U_1,U_2,...,U_{i-1}) - H(U_i|L_i,L_{i-1},U_{i-1})\\
\overset{(e)}{\leq}&H(U_i|U_{i-1}) - H(U_i|L_i,L_{i-1},U_{i-1})\\
=&I(L_i,L_{i-1};U_i|U_{i-1}),
\end{aligned}$$

where (c) follows from the chain rule of mutual information, (d) follows from that the second term equals to zero since we made the restriction on location releasing mechanism that U_i only depends on L_i, U_{i-1} and L_{i-1}, and (e) follows from the fact that conditioning does not increase entropy.

Since the objective function in $\mathcal{L}^*_{\text{online}}(D)$ is less than or equal to the one in $\mathcal{L}^{\text{Markov}}_{upper}(D)$, the constraint remains the same, and we also replace the probability distribution set $q(u_i|\boldsymbol{l}^i,\boldsymbol{u}^{i-1})$ with a smaller set $q(u_i|l_i,u_{i-1},l_{i-1})$, the proof of $\mathcal{L}^*_{\text{online}}(D) \leq \mathcal{L}^{\text{Markov}}_{upper}(D)$ is completed.

2) $\mathcal{L}^{Markov}_{lower}(D) \leq \mathcal{L}^*_{online}(D)$: Similar to the proof above, we show the lower bound of the objective function $I(\boldsymbol{L}^i;U_i|\boldsymbol{U}^{i-1})$ as the following.

$$\begin{aligned}
&I(\boldsymbol{L};U_i|\boldsymbol{U}^{i-1})\\
\overset{(f)}{=}&H(U_i|U_1,U_2,...,U_{i-1}) - H(U_i|U_{i-1},L_i,L_{i-1})\\
\overset{(g)}{\geq}&H(U_i|U_1,U_2,...,U_{i-1},L_{i-1}) - H(U_i|U_{i-1},L_i,L_{i-1})\\
\overset{(h)}{=}&H(U_i|U_{i-1},L_{i-1}) - H(U_i|U_{i-1},L_i,L_{i-1})\\
=&I(L_i;U_i|U_{i-1},L_{i-1}),
\end{aligned}$$

where (f) follows the same proofs as those in (c) and (d), (g) follows from the fact that conditioning does not increase entropy, and (h) follows from fact that U_i only depends on U_{i-1} and L_{i-1}. This is because we have the restriction on location releasing mechanism that U_i only depends on L_i, U_{i-1} and L_{i-1}, and L_i only depends on L_{i-1} according to the property of the first order markov transition in location traces, then the fact that U_i only depends on U_{i-1} and L_{i-1} is true, thus (h) holds.

Because the objective function in $\mathcal{L}^{\text{Markov}}_{lower}(D)$ is less than or equal to the one in $\mathcal{L}^{*}_{\text{online}}(D)$, and the constraint remains the same, the proof of $\mathcal{L}^{\text{Markov}}_{\text{lower}}(D) \leq \mathcal{L}^{*}_{\text{online}}(D)$ is completed.

Proofs in B1 and B2 complete the proof of Theorem 1.

C. Proof of Corollary 1

1) Proof of $\mathcal{L}^{Actual}_{online}(LPPM^*) \leq \mathcal{L}^{Markov}_{upper}(D)$: From the proof for $I(\boldsymbol{L}^i; U_i|\boldsymbol{U}^{i-1}) \leq I(L_i, L_{i-1}; U_i|U_{i-1})$ in Appendix B1, we can easily see that $\sum_{i=1}^{T} I(\boldsymbol{L}^i; U_i|\boldsymbol{U}^{i-1}) \leq \sum_{i=1}^{T} I(L_i, L_{i-1}; U_i|U_{i-1})$ is always true under the Markov release restriction. Therefore, when we use the same $q(u_i|l_i, u_{i-1}, l_{i-1})$ generated based on $\mathcal{L}^{\text{Markov}}_{\text{upper}}(D)$ to calculate the actual leakage $\sum_{i=1}^{T} I(\boldsymbol{L}^i; U_i|\boldsymbol{U}^{i-1})$, the actual leakage $\sum_{i=1}^{T} I(\boldsymbol{L}^i; U_i|\boldsymbol{U}^{i-1})$ is always less than or equal to $\mathcal{L}^{\text{Markov}}_{\text{upper}}(D)$ according to the proof in Appendix B1. That is to say, the actual leakage evaluated by $\sum_{i=1}^{T} I(\boldsymbol{L}^i; U_i|\boldsymbol{U}^{i-1})$ of the LPPMs generated based on $\mathcal{L}^{\text{Markov}}_{\text{upper}}(D)$ is upper bounded by $\mathcal{L}^{\text{Markov}}_{\text{upper}}(D)$.

2) Proof of $\mathcal{L}^{Markov}_{lower}(D) \leq \mathcal{L}^{Actual}_{online}(LPPM^*)$: Since we know that $\sum_{i=1}^{T} I(\boldsymbol{L}^i; U_i|\boldsymbol{U}^{i-1})$ is always greater or equal to $\mathcal{L}^{*}_{\text{online}}(D)$, and we also have $\mathcal{L}^{\text{Markov}}_{\text{lower}}(D) \leq \mathcal{L}^{*}_{\text{online}}(D)$, thus we have $\mathcal{L}^{\text{Markov}}_{\text{lower}}(D) \leq \sum_{i=1}^{T} I(\boldsymbol{L}^i; U_i|\boldsymbol{U}^{i-1})$.

Proofs in C1 and C2 complete the proof of Corollary 1.

D. Proof of Theorem 2

1) $\mathcal{L}^{*}_{online}(D) \leq \mathcal{L}_{upper}(D)$: The key idea of this proof is based on the fact that minimizing an objective function over a subset of certain constraint is less than or equal to minimizing it over the original constraint.

$$
\begin{aligned}
\mathcal{L}^{*}_{\text{online}}(D) &= \sum_{i=1}^{T} \min_{\substack{q(u_i|\boldsymbol{l}^i, \boldsymbol{u}^{i-1}): \\ \{D(L_i;U_i)\leq D_i\}_{i=1}^{T}}} I(\boldsymbol{L}^i; U_i|\boldsymbol{U}^{i-1}) \\
&\overset{(i)}{\leq} \sum_{i=1}^{T} \min_{\substack{q(u_i|l_i): \\ \{D(L_i;U_i)\leq D_i\}_{i=1}^{T}}} I(\boldsymbol{L}^i; U_i|\boldsymbol{U}^{i-1}) \\
&\overset{(j)}{\leq} \sum_{i=1}^{T} \min_{\substack{q(u_i|l_i): \\ \{D(L_i;U_i)\leq D_i\}_{i=1}^{T}}} \{I(L_i; U_i|\boldsymbol{U}^{i-1}) \\
&\quad + I(L_1, ...L_{i-1}; U_i|\boldsymbol{U}^{i-1}, L_i\} \\
&\overset{(k)}{=} \sum_{i=1}^{T} \min_{\substack{q(u_i|l_i): \\ \{D(L_i;U_i)\leq D_i\}_{i=1}^{T}}} I(L_i; U_i|\boldsymbol{U}^{i-1}) \\
&\overset{(l)}{=} \sum_{i=1}^{T} \min_{\substack{q(u_i|l_i): \\ \{D(L_i;U_i)\leq D_i\}_{i=1}^{T}}} I(L_i; U_i) = \mathcal{L}_{\text{upper}}(D),
\end{aligned}
$$

where (i) follows from the fact that $q(u_i|l_i)$ is a subset of $q(u_i|\boldsymbol{l}^i, \boldsymbol{u}^{i-1})$, (j) follows from the chain rule of conditional mutual information, (k) follows from the fact that $I(L_1, ...L_{i-1}; U_i|\boldsymbol{U}^{i-1}, L_i)$ equals to zero when choosing $q(u_i|l_i)$ in the minimization problem, and (l) follows from that fact that U^{i-1}, L_i, U_i forms a Markov chain (denoted by $\boldsymbol{U}^{i-1} \to L_i \to U_i$) when choosing $q(u_i|l_i)$.

2) $\mathcal{L}_{lower}(D) \leq \mathcal{L}^{*}_{offline}(D)$: We start with proving the lower bound on the objective function $I(\boldsymbol{L}; \boldsymbol{U})$ in $\mathcal{L}_{\text{lower}}(D)$.

$$
\begin{aligned}
&I(\boldsymbol{L}; \boldsymbol{U}) \\
&\overset{(m)}{=} I(L_i; \boldsymbol{U}) + I(L_1, ..., L_{i-1}, L_{i+1}, ..., L_T; \boldsymbol{U}|L_i) \\
&\overset{(n)}{=} I(L_i; U_i) + I(L_i; U_1, ..., U_{i-1}, U_{i+1}, ..., U_T|U_i) \\
&\quad + I(L_1, ..., L_{i-1}, L_{i+1}, ..., L_T; \boldsymbol{U}|L_i) \overset{(o)}{\geq} I(L_i; U_i)
\end{aligned}
$$

where (m) and (n) follow from the chain rule of mutual information, and (o) follows from the fact that mutual information is always nonnegative.

Then we have

$$
\begin{aligned}
\mathcal{L}^{*}_{\text{offline}}(D) &= \min_{q(\boldsymbol{u}|\boldsymbol{l}): \{D(L_i;U_i)\leq D_i\}_{i=1}^{T}} I(\boldsymbol{L}; \boldsymbol{U}) \\
&\overset{(p)}{\geq} T \cdot \min_{i} \min_{q(\boldsymbol{u}|\boldsymbol{l}): \{D(L_i;U_i)\leq D_i\}_{i=1}^{T}} I(L_i; U_i) \\
&\overset{(q)}{=} T \cdot \min_{i} \min_{q(u_i|l_i): D(L_i;U_i)\leq D_i} I(L_i; U_i) = \mathcal{L}_{\text{lower}}(D)
\end{aligned}
$$

Where (p) follows from the proof of $I(\boldsymbol{L}; \boldsymbol{U}) \geq I(L_i; U_i)$ for any integer i between 1 and T, (q) follows from the fact that the objective function $I(L_i; U_i)$ only depends on L_i and U_i, therefore, when choosing the conditional probability distribution as $q(u_i|l_i)$ rather than $q(\boldsymbol{u}|\boldsymbol{l})$, the results of the minimization problems remain the same.

Proofs in D1 and D2 complete the proof of Theorem 2.

ACKNOWLEDGEMENT

This work was supported by U.S. NSF CNS Grant 1731164, 1715947. Hui Li and part of Wenjing Zhang's work were supported by National Key Research and Development Program of China (2017YFB0802200), NSFC 61732022, U1401251, Shaanxi nature science research project 2016ZDJC-04, 111 Project B16037. This work was done when Wenjing Zhang was a visiting Ph.D. student in the department of ECE at the University of Arizona.

REFERENCES

[1] I. A. Junglas and R. T. Watson, "Location-based services," *Communications of the ACM*, vol. 51, no. 3, pp. 65–69, 2008.

[2] A. Dey, J. Hightower, E. de Lara, and N. Davies, "Location-based services," *IEEE Pervasive Computing*, vol. 9, no. 1, pp. 11–12, 2010.

[3] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: user movement in location-based social networks," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011, pp. 1082–1090.

[4] S. Eubank, H. Guclu, V. A. Kumar, M. V. Marathe *et al.*, "Modelling disease outbreaks in realistic urban social networks," *Nature*, vol. 429, no. 6988, pp. 180–184, 2004.

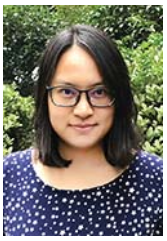
[5] S. Dhar and U. Varshney, "Challenges and business models for mobile location-based services and advertising," *Communications of the ACM*, vol. 54, no. 5, pp. 121–128, 2011.

[6] C.-Y. Chow and M. F. Mokbel, "Trajectory privacy in location-based services and data publication," *ACM SIGKDD Explorations Newsletter*, vol. 13, no. 1, pp. 19–29, 2011.

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

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

[9] R. Shokri, G. Theodorakopoulos, C. Troncoso, J.-P. Hubaux, and J.-Y. Le Boudec, "Protecting location privacy: optimal strategy against localization attacks," in *Proceedings of the 19th ACM conference on Computer and Communications Security*. ACM, 2012, pp. 617–627.
[10] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," in *Proceedings of the 20th ACM conference on Computer and Communications Security*. ACM, 2013, pp. 901–914.
[11] S. Oya, C. Troncoso, and F. Pérez-González, "Back to the drawing board: Revisiting the design of optimal location privacy-preserving mechanisms," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 1959–1972.
[12] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.
[13] C. Dwork, "Differential privacy," in *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming*, 2006, pp. 1–12.
[14] R. Shokri, C. Troncoso, C. Diaz, J. Freudiger, and J.-P. Hubaux, "Unraveling an old cloak: k-anonymity for location privacy," in *Proceedings of the 9th annual ACM workshop on Privacy in the electronic society*. ACM, 2010, pp. 115–118.
[15] S. Mascetti, D. Freni, C. Bettini, X. S. Wang, and S. Jajodia, "Privacy in geo-social networks: proximity notification with untrusted service providers and curious buddies," *The VLDB JournalThe International Journal on Very Large Data Bases*, vol. 20, no. 4, pp. 541–566, 2011.
[16] C.-Y. Chow, M. F. Mokbel, and X. Liu, "Spatial cloaking for anonymous location-based services in mobile peer-to-peer environments," *GeoInformatica*, vol. 15, no. 2, pp. 351–380, 2011.
[17] Y. Xiao and L. Xiong, "Protecting locations with differential privacy under temporal correlations," in *Proceedings of the 22nd ACM Conference on Computer and Communications Security*. ACM, 2015, pp. 1298–1309.
[18] T. Murakami, "Expectation-maximization tensor factorization for practical location privacy attacks," *in Proceedings on Privacy Enhancing Technologies*, vol. 4, pp. 58–75, 2017.
[19] J. Krumm, "Inference attacks on location tracks," *Lecture Notes in Computer Science*, vol. 4480, pp. 127–143, 2007.
[20] A. Eland, "Tackling urban mobility with technology," https://europe.googleblog.com/2015/11/tackling-urban-mobility-with-technology.html, 2015.
[21] 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 *Proceedings of the 13th Workshop on Privacy in the Electronic Society*. ACM, 2014, pp. 73–82.
[22] R. Shokri, G. Theodorakopoulos, and C. Troncoso, "Privacy games along location traces: A game-theoretic framework for optimizing location privacy," *ACM Transactions on Privacy and Security*, vol. 19, no. 4, pp. 11:1–11:31, 2016.
[23] ——, "Privacy games along location traces: A game-theoretic framework for optimizing location privacy," *ACM Transactions on Privacy and Security (TOPS)*, vol. 19, no. 4, p. 11, 2017.
[24] C. Y. Ma and D. K. Yau, "On information-theoretic measures for quantifying privacy protection of time-series data," in *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*. ACM, 2015, pp. 427–438.
[25] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux, "Quantifying location privacy," in *2011 IEEE Symposium on Security and Privacy*. IEEE, 2011, pp. 247–262.
[26] R. Shokri, G. Theodorakopoulos, G. Danezis, J.-P. Hubaux, and J.-Y. Le Boudec, "Quantifying location privacy: the case of sporadic location exposure," in *Proceedings of the 11th International Conference on Privacy Enhancing Technologies*. Springer, 2011, pp. 57–76.
[27] P. Cuff and L. Yu, "Differential privacy as a mutual information constraint," in *Proceedings of the 23rd ACM Conference on Computer and Communications Security*. ACM, 2016, pp. 43–54.
[28] L. Sankar, S. R. Rajagopalan, and H. V. Poor, "Utility-privacy tradeoffs in databases: An information-theoretic approach," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 6, pp. 838–852, 2013.
[29] F. du Pin Calmon and N. Fawaz, "Privacy against statistical inference," in *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2012, pp. 1401–1408.
[30] S. Salamatian, A. Zhang, F. du Pin Calmon, S. Bhamidipati, N. Fawaz, B. Kveton, P. Oliveira, and N. Taft, "How to hide the elephant-or the donkey-in the room: Practical privacy against statistical inference for large data." in *2013 IEEE Global Conference on Signal and Information Processing*, 2013, pp. 269–272.
[31] R. Blahut, "Computation of channel capacity and rate-distortion functions," *IEEE Transactions on Information Theory*, vol. 18, no. 4, pp. 460–473, 1972.
[32] M. Götz, S. Nath, and J. Gehrke, "Maskit: Privately releasing user context streams for personalized mobile applications," in *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. ACM, 2012, pp. 289–300.
[33] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley & Sons, 2012.
[34] I. Csisz and G. Tusnády, "Information geometry and alternating minimization procedures," *Statistics and Decisions*, pp. 205–237, 1984.
[35] Y. Zheng, X. Xie, and W.-Y. Ma, "Geolife: A collaborative social networking service among user, location and trajectory." *IEEE Data Engineering Bulletin*, vol. 33, no. 2, pp. 32–39, 2010.
[36] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.



Wenjing Zhang is a Ph.D. student at the School of Cyber Engineering, Xidian University, Xi'an, China. She received her B.S. degree in information security and M.S. degree in communication and information system from Xidian University in 2011 and 2014. She was a visiting Ph.D. student in the Department of Electrical and Computer Engineering at the University of Arizona from 2016 to 2018. Her current research interests focus on data privacy and privacy metrics. She is a student member of IEEE.

Ming Li is an Associate Professor in the Department of Electrical and Computer Engineering of University of Arizona. He was an Assistant Professor in the Computer Science Department at Utah State University from 2011 to 2015. He received his Ph.D. in ECE from Worcester Polytechnic Institute in 2011. His main research interests are wireless networks and security, with current emphases on wireless network optimization, wireless security and privacy, and cyber-physical system security. He received the NSF Early Faculty Development (CAREER) Award in 2014, and the ONR Young Investigator Program (YIP) Award in 2016. He is a senior member of IEEE and a member of ACM.

Ravi Tandon is an Assistant Professor in the Department of Electrical and Computer Engineering at the University of Arizona. Prior to joining the University of Arizona in Fall 2015, he was a Research Assistant Professor at Virginia Tech with positions in the Bradley Department of ECE, Hume Center for National Security and Technology and at the Discovery Analytics Center in the Department of Computer Science. He received the B.Tech. degree in Electrical Engineering from the Indian Institute of Technology, Kanpur in 2004 and the Ph.D. degree in Electrical and Computer Engineering from the University of Maryland, College Park (UMCP) in 2010. From 2010 to 2012, he was a postdoctoral research associate in the Department of Electrical Engineering at Princeton University. He is a recipient of the Best Paper Award at IEEE GLOBECOM 2011. He was nominated for the Graduate School Best Dissertation Award, and also for the ECE Distinguished Dissertation Fellowship Award at the University of Maryland, College Park. His current research interests include information theory and its applications to wireless networks, communications, security and privacy distributed storage systems, machine learning and data mining. He is a senior member of IEEE.

Hui Li is a Professor at the School of Cyber Engineering, Xidian University, Xi'an, China. He received B.Sc. degree from Fudan University in 1990, M.Sc. and Ph.D. degrees from Xidian University in 1993 and 1998. In 2009, he was with Department of ECE, University of Waterloo as a visiting scholar. His research interests are in the areas of cryptography, security of cloud computing, wireless network security, information theory. He is a member of IEEE.