Protecting Spatiotemporal Event Privacy in Continuous Location-Based Services

Yang Cao, Yonghui Xiao, Li Xiong, Liquan Bai, and Masatoshi Yoshikawa

Abstract—Location privacy-preserving mechanisms (LPPMs) have been extensively studied for protecting users' location privacy by releasing a perturbed location to third parties such as location-based service providers. However, when a user's perturbed locations are released continuously, existing LPPMs may not protect the sensitive information about the user's real-world activities, such as "visited hospital in the last week" or "regularly commuting between location A and location B every weekday" (it is easy to infer that location A and location B may be home and office), which we call it *spatiotemporal event*. In this paper, we first formally define spatiotemporal event as Boolean expressions between location and time predicates, and then we define ϵ-*spatiotemporal event privacy* by extending the notion of differential privacy. Second, to understand how much spatiotemporal event privacy that existing LPPMs can provide, we design computationally efficient algorithms to quantify the spatiotemporal event privacy leakage of state-of-the-art LPPMs. It turns out that the existing LPPMs may not adequately protect spatiotemporal event privacy. Third, we propose a framework, PriSTE, to transform an existing LPPM into one protecting spatiotemporal event privacy by calibrating the LPPM's privacy budgets. Our experiments on real-life and synthetic data verified that the proposed method is effective and efficient.

Index Terms—Location-based services, location privacy, location obfuscation, Markov model, trajectory privacy

1 Introduction

The continued advances and usage of smartphones and GPS-enabled devices have provided tremendous opportunities for Location-Based Service (LBS), such as Yelp or Uber for snapshot or continuous queries, for example, "where is the nearest restaurant" or "continuously report the taxis within one mile of my location". Mobile users have to share their real-time locations or a sequence of locations with the service providers, which raises privacy concerns since users' digital trace can be used to infer sensitive information, such as home and workplace, religious places and sexual inclinations [2], [3], [4].

A large number of studies (see surveys [5], [6], [7]) have explored how to protect user's location privacy which can be categorized from different aspects: privacy goals, adversarial models, location privacy metrics, and location privacy preserving mechanisms (LPPMs). *Privacy goals* indicate what should be protected or what are the secrets (e.g., a single location or a trajectory); *adversarial models* make assumptions about the adversaries; *location privacy metrics* formally define the quantitative measurement of the protection w.r.t. the privacy goal; *LPPMs* are designed to achieve a specified privacy metrics. For instance, Geo-Indistinguishability [8] is a location privacy metrics, which is receiving increasing attention since it extends the notion of differential privacy [9] to the location privacy setting so that the protection level does not depend on adversaries' prior knowledge; the privacy goal of Geo-Indistinguishability is to protect a single location (can be extended for protecting location trace [10]); Laplace Planar Mechanism [8] is an LPPM satisfying Geo-Indistinguishability. Another example is Planar Isotropic Mechanism [11] for the metrics of δ-location set privacy to protect each location in a trajectory. These state-of-the-art LPPMs take an actual location and a privacy parameter as inputs and probabilistically output a randomly perturbed location. A LPPM privacy parameter controls the location privacy level. For examples of the above mechanisms, the privacy parameter is dened as a positive real value and a smaller privacy parameter indicates stronger privacy protection. In other words, the privacy parameter can be considered as the controlled level of privacy loss.

We argue that the existing techniques may not adequately protect users' sensitive information in their real-world activities because the *privacy goal* is not well-defined. Most of the existing studies only focused on the protection of either a single location or a trajectory, which does not completely reflect the secrets that should be protected in users' real-world activities. To explain this, we need to formally define the sensitive information in the users' real-world activities. We define a user's a single location at time t as a predicate $l_t = s_i$ where l_t is a variable representing the user's position at time t and $s_i \in \mathbb{S}, i \in [1, m]$ is a location on the map $\mathbb{S}$ of m locations. The value of such predicate can be either *true* or *false*, which could be a secret of the user. Then, we can generalize users' secrets in their real-world activities as Boolean expressions of combining different predicates

- Y. Cao and M. Yoshikawa are with the Department of Social Informatics, Kyoto University, Kyoto 606-8501, Japan. E-mail: {yang, yoshikawa}@i.kyoto-u.ac.jp.
- Y. Xiao is with the Google Inc., Mountain View, CA 94043. E-mail: yohu@google.com.
- L. Xiong and L. Bai are with the Department of Computer Science, Emory University, Atlanta, GA 30322. E-mail: {lxiong, liquan.bai}@emory.edu.

Manuscript received 24 July 2019; revised 11 Nov. 2019; accepted 12 Dec. 2019. Date of publication 0 . 0000; date of current version 0 . 0000. (Corresponding author: Yang Cao.) Recommended for acceptance by J. Xu. Digital Object Identifier no. 10.1109/TKDE.2019.2963312

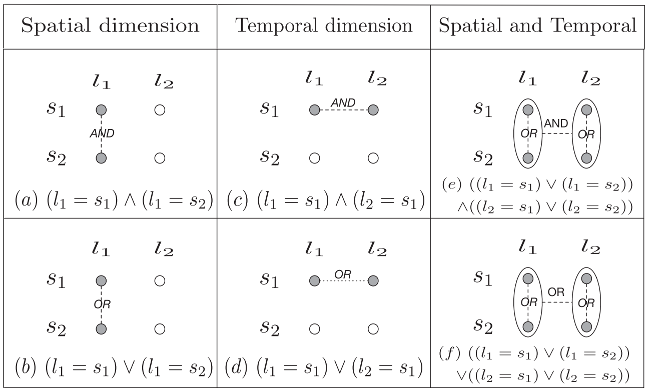


Fig. 1. Six examples of spatiotemporal events. Event (a) is always false. Event (b) represents a sensitive *region*. Event (c) represents a sensitive *trajectory*. Event (d) represents the *presence* or not in a sensitive location. Event (e) indicates a mobility *pattern* passing through sensitive regions. Event (f) indicates the *presence* or not in a sensitive region.

over spatial and/or temporal dimensions, which is called *spatiotemporal event* in this paper.

In Fig. 1, we illustrate six representative Boolean expressions between location and time dimensions. We use s_1 and s_2 to denote two locations on the map $\mathbb{S}$, and use l_1 and l_2 to denote two variables about a user's locations at timestamps 1 and 2, respectively. Event (a) is always false since a user cannot be physically at two different locations at the same time. Event (b) means that the secret is a sensitive region (or area) of $\{s_1, s_2\}$. Event (c) represents a sensitive trajectory $s_1 \rightarrow s_1$ between timestamps 1 and 2, i.e., the user stays at s_1 at time 1 and time 2. Event (d) denotes that the secret is the visit to s_1 at timestamp 1 or 2. Event (e) depicts the secret as a type of trajectory *pattern*, i.e., the user may stay at two sensitive regions successively; a real-world example of such event is "regularly commuting between Address 1 and Address 2 every morning and every afternoon", i.e., periodic spatiotemporal events may happen every week day. Event (f) indicates the secret as user's *presence* in sensitive region $\{s_1, s_2\}$ at either timestamp 1 or 2; a real-world example of such event is "visited hospital in the last week", i.e., the hospital visit may happen once or multiple times at any time in last week.

We can see that the spatiotemporal events representing sensitive locations and a trajectory (i.e., (b) and (c)), which are the major privacy goals of previous studies, are only two cases among the six enumerated examples. Hence, even if an LPPM protects each location or a trajectory, it may not protect a complex spatiotemporal event such as the ones shown in Figs. 1e and 1f since such new privacy goals have not been formalized in the literature.

In this paper, we attempt to achieve spatiotemporal event privacy by leveraging the existing LPPMs designed for conventional location privacy. There are three major challenges below. First, we lack the formal definition of spatiotemporal event and privacy metrics for it. Second, evaluating the privacy guarantee of a given spatiotemporal event could be computationally intractable since the event can be extremely complicated. Taking the *pattern* event (e.g., Fig. 1e) for example, if the sensitive region includes m locations and the length of such event spans T timestamps, there are m^T possible trajectories that need to be protected, which may lead to exponential time computation. Third, similar to Geo-Indistinguishability, we hope to design a mechanism that is robust to adversaries with *any* prior knowledge.

Contributions. To the best of our knowledge, this is the first paper that studies how to achieve spatiotemporal event privacy. Our contributions are summarized as follows.

First, we study the privacy goal and privacy metrics for protecting spatiotemporal event (Section 2). We formally define a new type of privacy goal, i.e., spatiotemporal events, as Boolean expressions of location-time predicates, and propose a privacy metric, *ϵ-spatiotemporal event privacy*, for protecting spatiotemporal events by extending the notion of differential privacy. We also explore the difference between the metrics of location privacy and spatiotemporal event privacy. It turns out that, although the definition of spatiotemporal event is more general than a single location or a trajectory, the privacy metrics between spatiotemporal event privacy and location privacy can be orthogonal and complementary: Location privacy provides general protection against unknown risks, while spatiotemporal event privacy guarantees flexible and customizable protection which may not be provided by the existing LPPMs. Hence, it would be preferable that an LPPM achieving a location privacy metrics such as Geo-Indistinguishability can also satisfy ϵ-spatiotemporal event privacy w.r.t. user-specified events.

Second, we develop efficient algorithms for quantifying how much ϵ-spatiotemporal event privacy a given LPPM can provide w.r.t. adversaries with a specific prior knowledge about the user's initial probability distribution over possible locations (Section 3). We model an LPPM as an emission matrix that takes user's true position as input and outputs a perturbed location. As we mentioned previously, one of the challenges in quantifying the probability of a spatiotemporal event is that the computational complexity may be exponentially increasing with the number of predicates in a user-specified spatiotemporal event. We develop a novel *two-possible-world* method to quantify spatiotemporal event privacy with linear complexity.

Third, based on our quantification method, we propose a framework, i.e., PriSTE (Private Spatio-Temporal Event), which converts a mechanism for location privacy into one for spatiotemporal event privacy against adversaries with any prior knowledge (Section 4). We demonstrate the effectiveness of our framework by two case studies using state-of-the-art LPPMs, i.e., Laplace Planar Mechanism for Geo-Indistinguishability [8] and Planar Isotropic Mechanism for δ-location set privacy [11].

Finally, we evaluate our algorithms on both synthetic and real-world datasets testing its feasibility, efficiency, and the impact of various parameters (Section 5).

2 Defining Spatiotemporal Event Privacy

2.1 Scenario

We consider a scenario that a single user continuously releases her perturbed location with an untrusted third party such as a location-based service provider. The user's true locations are denoted by $l_1, l_2, \ldots, l_T$. A location privacy-preserving mechanism (LPPM) blurs user's true location l_t to a perturbed one o_t that satisfies a privacy metrics such as *Geo-Indistinguishability*[8] or *δ-location set privacy*

(a) Emission Matrix

perturbed location o_t

true loc l_t	s_1	s_2	s_3
s_1	0.5	0.3	0.2
s_2	0.1	0.8	0.1
s_3	0.2	0.2	0.6

$\Pr(o_t \mid l_t)$

(b) Transition Matrix

l_{t+1}

l_t	s_1	s_2	s_3
s_1	0.1	0.2	0.7
s_2	0	0	1
s_3	0.3	0.3	0.4

$\Pr(l_{t+1} \mid l_t)$

Fig. 2. Illustration of emission matrix and transition matrix.

[11]. Essentially, the LPPM is an emission matrix that takes user's true location as input and outputs a perturbed one.

We clarify our assumptions about LPPM and users' mobility model as follows. First, we consider an LPPM that takes input as user's true location l_t and outputs a perturbed location o_t at time t. We use an $m \times m$ emission matrix where each cell is the emission probability (as shown in Fig. 2a) to represent the LPPM. Second, we assume a user's location at time $t+1$ is correlated with her location at time t, representing by an $m \times m$ transition matrix as shown in Fig. 2b, and such transition matrix is public information which can be learned from either historical trajectory or the pattern of road networks. We model the correlation between user's consecutive locations using first-order[1] time-homogeneous[2] Markov model, i.e., the transition matrix is identical at each t. The transition matrix is given in our system. The major notations are summarized in Table 1.

2.2 Spatiotemporal Events

We first define location-time predicate, which is an atomic element in spatiotemporal events. Let $\mathbb{S} = \{s_1, \ldots, s_m\}$ be the domain of all possible locations, where m is the size of the domain and s_i is one location (we use *state* interchangeably) on the map. At time t, a user's location can be stated as $l_t = s_i$, which means the user is at location s_i at time t. We call $l_t = s_i$ *location-time predicate*, whose value can be *true* or *false* depending on the ground truth of user's state at t.

We define spatiotemporal events as Boolean expressions of the location-time predicates.

Definition 2.1 (EVENT). *A spatiotemporal event, denoted by* EVENT, *is a single location-time predicate or a combination of location-time predicates linked by the Boolean operators AND, OR, NOT (i.e.,* $\wedge, \vee, \neg$*, respectively).*

For the ease of exposition, we define the following notations. We denote a region (i.e., a set of locations) by a vector $\mathbf{s} \in \{0,1\}^{m\times 1}$ where the ith element is 1 only if the region contains s_i. We use $\mathcal{S}$ to indicate a sequence of regions. We denote the corresponding timestamp of each region by $\mathcal{T}$ as a sequence of timestamps with the same cardinality of $\mathcal{S}$.

Using Boolean logic to define spatiotemporal events enables users to customize their privacy preference for diverse real-world activities as shown in Fig. 1. A pair of ith

1. If the Markov model is high-ordered, i.e., the transition matrix has a larger state domain, our approach still works.

2. If the Markov model is time-varying, i.e., transition matrices at different t are not identical, our approach still works. We explain this in the next section.

TABLE 1
Notations

$\mathbb{S}$	Domain of the states, $\mathbb{S} = \{s_1, s_2, \ldots, s_m\}$
s_i, s_j, s_k	variables of the states, i.e., $s_i, s_j, s_k \in \mathbb{S}$
m	the amount of all possible locations on the map
$\mathbf{s}$	a vector representing a region, $\mathbf{s} \in \{0,1\}^{m\times 1}$
t	a timestamp in $\{1, 2, \ldots, T\}$
$\mathcal{S}$	a sequence of regions
$\mathcal{T}$	a sequence of timestamps
l_t	a user's true location at time t
o_t	a user's perturbed location at time t
EVENT	a spatiotemporal event
$\tilde{\mathbf{p}}_{o_t}$	emission probabilities given the observation o_t.
$\tilde{\mathbf{p}}^{\mathbf{D}}_{o_t}$	a diagonal matrix with the vector $\tilde{\mathbf{p}}_{o_t}$ on the diagonal.
π	initial probability $\boldsymbol{\pi} \in \mathbb{R}^{1\times m}$

elements in $\mathcal{S}$ and $\mathcal{T}$ could form a *single region event* as shown in Event (b) in Fig. 1. These single region events could be combined by AND or OR, which form PRESENCE or PATTERN (e.g., Events (e) and (f) in Fig. 1).

2.2.1 PRESENCE Event

When the secret is whether or not a user visited a sensitive region (e.g., medical facilities) in a given time period, we can use PRESENCE to represent such secret. A PRESENCE event holds if a user appears in *any* one of the regions with user-specified timestamps. In the simplest case of PRESENCE, when the region includes only one location and the time period consists of one timestamp, it reduces to a *single location event*. Hence, PRESENCE event can be seen as a generalization of single location event.

Definition 2.2 (PRESENCE). *Given a sequence of regions* $\mathcal{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_n]$ *and a sequence of timestamps* $\mathcal{T} = [t_1, \ldots, t_n]$*, if a user appears in at least one* $\mathbf{s}_k \in \mathcal{S}$ *at the corresponding time* $t_k \in \mathcal{T}$*, then it is a presence event, denoted by* PRESENCE$(\mathcal{S}, \mathcal{T})$.

Example 2.1 (Example of PRESENCE). Fig. 3 shows a map of $\mathbb{S} = \{s_1, s_2, s_3\}$. For this event, the region $\mathbf{s} = [1,1,0]^\intercal$ denoting the states s_1 and s_2; the time period $\mathcal{T} = [3,4]$ denoting timestamp 3 and 4. Let $\mathcal{S} = [\mathbf{s}, \mathbf{s}]$. This PRESENCE $(\mathcal{S}, \mathcal{T})$ event is expressed as $(l_3 = s_1) \vee (l_3 = s_2) \vee (l_4 = s_1)$ $\vee (l_4 = s_2)$. The shaded region shows a PRESENCE event that the user appears in a region of $\{s_1, s_2\}$ during

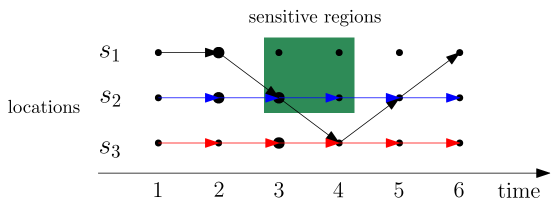


Fig. 3. We show two events, i.e., PRESENCE$(\mathcal{S}, \mathcal{T})$ and PATTERN$(\mathcal{S}, \mathcal{T})$. If the user's ground truth trajectory is the black one, only PRESENCE$(\mathcal{S}, \mathcal{T})$ is true; if the user's trajectory is the blue one, both events are true; if the user's trajectory is the red one, both events are false.

timestamps 3 and 4. If the user's true trajectory passes through the shaded region (at least one timestamp), the event is true.

2.2.2 Pattern Event

We use Pattern to represent the secret whether or not a user visited multiple sensitive regions successively. In a simple case of Pattern event, the regions consist of locations at a sequence of timestamps, then it is reduced to *single trajectory event*. Hence, Pattern is a generalization of a user's trajectories.

Definition 2.3 (Pattern). *Given a sequence of regions $\mathcal{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_n]$ and a sequence of timestamps $\mathcal{T} = [t_1, \ldots, t_n]$, if a user appears in all $\{\mathbf{s}_1, \ldots, \mathbf{s}_n\}$ sequentially at the corresponding time during $\mathcal{T}$, then it is a pattern event, denoted by* $\textsc{Pattern}(\mathcal{S}, \mathcal{T})$.

Example 2.2 (Example of Pattern). The Pattern event in Fig. 3 represents trajectories with a pattern going through a sensitive region $\{s_1, s_2\}$ at timestamp 2 and the same region $\{s_1, s_2\}$ at timestamp 3 successively. This Pattern event is expressed as $((l_2 = s_1) \vee (l_2 = s_2)) \wedge ((l_3 = s_1) \vee (l_3 = s_2))$.

2.2.3 Discussion

From the above definitions, we can see that, in terms of privacy goal, spatiotemporal event privacy can be a *generalization* of location privacy studied in the literature in which the privacy goal is protecting a single location or a trajectory. In this paper, we focus on the two representative events defined above, i.e., Presence and Pattern, which are the two most complicated and unexplored events among examples in Fig. 1. We note that Presence and Pattern include the cases when the time $\mathcal{T}$ is not consecutive. Users can specify one or multiple events to be protected.

On the other hand, it could be a non-trivial task for end-users to define a spatiotemporal event that needs to be protected. We provided a tool in our recent demonstration [12] to help users explore how accurate an adversary could infer a given event so that to identify and protect risky spatiotemporal events. Boolean logic is an expressive tool for representing spatiotemporal events which could be complicated. Besides the users burden on defining privacy preference, another negative effect (due to the expressiveness) may be that an event with complicated logic could be hard to protect with meaningful utility and reasonable runtime. We address the problem of computation complexity in Section 3.

2.3 ϵ-Spatiotemporal Event Privacy

Inspired by the definition of differential privacy [9], we define ϵ-Spatiotemporal Event Privacy as follows.

Definition 2.4 (ϵ-Spatiotemporal Event Privacy). *A mechanism preserves ϵ-Spatiotemporal Event Privacy for a spatiotemporal* Event *if at any timestamp t in $\{1, \ldots, T\}$ given any observations $\{o_1, \ldots, o_t\}$*

$$\Pr(o_1, \ldots, o_t | \textsc{Event}) \leq e^{\epsilon} \Pr(o_1, \ldots, o_t | \neg\textsc{Event}), \tag{1}$$

where Event *is a logic variable about the user-specified spatiotemporal event and $\neg$*Event *denotes the negation of* Event. $\Pr(o_1, o_2, \ldots, o_t | \textsc{Event})$ *denotes the probability of the observations $o_1, o_2, \ldots, o_t$ given the value of* Event.

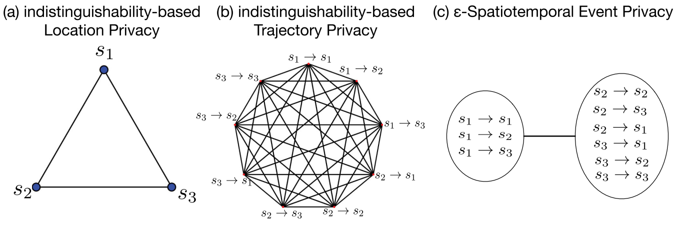


Fig. 4. Illustration of indistinguishability-based privacy metrics for distinct privacy goals when $\mathcal{S} = \{s_1, s_2, s_3\}$ and $\mathcal{T} = 2$.

There are two major benefits of extending differential privacy to protecting spatiotemporal events. First, it provides a well-defined semantics for spatiotemporal event privacy. Similar to differential privacy that requires the indistinguishability between any two neighboring databases[9], ϵ-Spatiotemporal Event Privacy requires the indistinguishability regarding whether the Event is true or false given any observations. It provides a clear privacy semantics: it is hard for adversaries to distinguish whether the event happened or not. Another benefit is that, similar to differential privacy whose privacy guarantee is independent of the prior probability of a given database, the privacy provided by ϵ-Spatiotemporal Event Privacy is independent of the prior probability of the protected event.

To better understand the characteristics of spatiotemporal event privacy, we illustrate the indistinguishability-based privacy metrics for the three privacy goals in Fig. 4, where the lines connecting two secrets indicate the requirements of indistinguishability between the corresponding two possible values of the secrets.

As shown in Fig. 4a, indistinguishability-based location privacy metrics (such as Geo-Indistinguishability[8]) requires indistinguishability between each pair of locations. Indistinguishability-based trajectory privacy metrics [10], [11], [13] requires indistinguishability between each pair of possible trajectories as shown in Fig. 4b. Whereas, ϵ-spatiotemporal event privacy requires indistinguishability between the defined event and its negation. For example, if the spatiotemporal event is defined as $\textsc{Pattern}(\mathcal{S}, \mathcal{T})$ where $\mathcal{S} = [\mathbf{s}_1, \mathbf{s}_2], \mathbf{s}_1 = \{s_1\}, \mathbf{s}_2 = \{s_1, s_2, s_3\}$ and $\mathcal{T} = [1, 2]$ (i.e., a trajectory passes through s_1 and then a region $\{s_1, s_2, s_3\}$ successively), then it only requires the indistinguishability between the set of all possible trajectories that pass through $\{s_1\}$ and $\{s_1, s_2, s_3\}$ and the set of trajectories that do not. This spatiotemporal event privacy makes sense when some mobility patterns are sensitive. For example, if s_1 is "hospital", s_2 is "home", and s_3 is "office", the pattern from s_1 to $\{s_1, s_2, s_3\}$ could be sensitive.

We note that spatiotemporal event privacy is orthogonal to location privacy or trajectory privacy. First, protecting the privacy of a single location or a trajectory may not imply the protection of spatiotemporal event privacy because spatiotemporal event can be complex as shown in Figs. 1e or 1f. The existing LPPMs are designed to ensure privacy metrics defined on locations or trajectories. One of our focus in this study is to quantify how much spatiotemporal event privacy a given LPPM can provide, which will be elaborated in the next section. Second, protecting spatiotemporal event privacy does not imply the protection of location privacy because they define indistinguishability over different level of secrets. Taking Fig. 4c for example, the indistinguishability between $s_1 \rightarrow s_1$ and $s_1 \rightarrow s_2$ is not required in such spatiotemporal event privacy guarantee; however, it is required in trajectory privacy as shown in Fig. 4b. Even if we define the event in spatiotemporal event privacy as a single location, say s_1, the guarantee of spatiotemporal event privacy is the indistinguishability between s_1 and $\{s_2, s_3\}$, which does not guarantee the indistinguishability between s_1 and s_2.

It would be preferable if we achieve both location privacy and spatiotemporal event privacy so that a user can enjoy the best of two worlds: Location privacy provides general protection against unknown risks when sharing location with the third parties, while spatiotemporal event privacy guarantees customizable protection which may prevent against profiling attacks [3], [14]. Therefore, in this paper, we study how to use an existing probabilistic LPPM (e.g., Laplace Planar Mechanism [8] and Planar Isotropic Mechanism [11]) to achieve ϵ-spatiotemporal event privacy.

We note that the definition of events may reveal a user's sensitive information. In this paper, we assume that the events and the protection mechanisms are locally and securely stored in the user's device. The user may specify one or multiple events that need to be protected. In practice, we can also have default that are suggested by a privacy preference recommendation system for users' selection [15] or pre-specified event templates that are given by the user.

3 Quantifying Spatiotemporal Event Privacy

3.1 Overview of Our Approach

For ease of exposition, we first assume that adversaries who have a specific knowledge of the user's initial probability distribution over possible locations, which is denoted by $\boldsymbol{\pi}$; in the next section, we will remove this assumption so that the spatiotemporal event privacy leakage will be bounded in ϵ w.r.t. adversaries with any prior knowledge of user's initial probability.

Now, we explain the main goal of quantifying the spatiotemporal event privacy leakage of the LPPM and our approach. Based on Definition 2.4 of ϵ-spatiotemporal event privacy, we need to calculate the maximum ratio of $\frac{\Pr(o_1,o_2,\ldots,o_T|\textsc{Event})}{\Pr(o_1,o_2,\ldots,o_T|\neg\textsc{Event})}$ in which $o_1, o_2, \ldots, o_T$ are released by a given LPPM. This ratio can be considered as spatiotemporal event privacy leakage w.r.t. the user-specified event. We quantify this ratio w.r.t. given observations $o_1, o_2, \ldots, o_T$ and a given user's initial probability $\boldsymbol{\pi}$, so that we can directly calculate the $\Pr(o_1, o_2, \ldots, o_T|\textsc{Event})$. In Section 4, we will design a mechanism for spatiotemporal event privacy w.r.t. any observations and arbitrary initial probability. Our goal in this section is to calculate the likelihood of the observations given Event or ¬Event , i.e., $\Pr(o_1, o_2, \ldots, o_T|\textsc{Event})$ or $\Pr(o_1, o_2, \ldots, o_T|\neg\textsc{Event})$, which can be derived by $\Pr(o_1, o_2, \ldots, o_T|\textsc{Event}) = \frac{\Pr(o_1,o_2,\ldots,o_T,\textsc{Event})}{\Pr(\textsc{Event})}$. We call $\Pr(\textsc{Event})$ as *prior probability* of the event, and $\Pr(o_1, o_2, \ldots, o_T, \textsc{Event})$ as *joint probability* of the event.

A severe challenge of calculating the prior or joint probabilities of the event is the computational complexity. Given an arbitrary spatiotemporal event, we need to enumerate all possible combination of the Boolean expression for prior and joint probabilities, which can be exponential to the number of predicates in the expression. To address this problem, we propose a two-possible-world method for computing the prior and joint probabilities in Sections 3.2 and 3.3.

For ease of exposition, we define notations below. $\mathbf{M} \in \mathbb{R}^{m\times m}$ denotes a transition matrix that describes temporal correlations in user's location. At timestamp 1, an initial probability is denoted by $\boldsymbol{\pi} \in [0,1]^{1\times m}$. During timestamp $\{1, 2, \ldots, T\}$, the probability of the true location $\Pr(l_t)$ is denoted by a row vector $\mathbf{p}_t \in [0,1]^{1\times m}$ where ith element denotes $\Pr(l_t = s_i)$. A Markov model follows the transition property of $\mathbf{p}_{t+1} = \mathbf{p}_t\mathbf{M}$, e.g., after a Markov transition, $\mathbf{p}_2 = \boldsymbol{\pi}\mathbf{M}$ at timestamp 2 given $\mathbf{p}_1 = \boldsymbol{\pi}$.

The notations below for matrix computation are also used in the rest of this paper. Let $\mathbf{0}$ and $\mathbf{1}$ be row vectors with m elements being 0 and 1 respectively. $[\mathbf{0}, \mathbf{1}]$ is a row vector in $\mathbb{R}^{1\times 2m}$. $\mathbf{a} \circ \mathbf{b}$ denotes the Hadamard product of $\mathbf{a}$ and $\mathbf{b}$. $\mathbf{a}^{\mathbf{D}}$ is a diagonal matrix with the elements of vector $\mathbf{a}$ on the diagonal.

3.2 Computing Prior Probability of an Event

To avoid the exponential complexity, we propose an efficient algorithm with two possible worlds. The idea is to elaborate a "new" transition matrix $\mathbf{M}_t \in \mathbb{R}^{2m\times 2m}$ at each time t which encodes the complex spatiotemporal event inside, so that the calculation of the prior or joint probability for a complicated event is the same as one simple predicate.

Intuition. The main idea of our method is to use two virtual worlds denoting whether the Event is true or false. The states in the two worlds denote the joint probabilities $\Pr(l_t = s_i, \textsc{Event})$ and $\Pr(l_t = s_i, \neg\textsc{Event})$. For Presence, once a trajectory enters into the region of the Presence, its probability will be kept in the world of true Event forever. For Pattern, the probability distribution among the two worlds are derived at the beginning timestamp of the Event, and only the trajectories satisfying the Pattern will be kept in the world of true Event. At last, the sum of probabilities in the world of true Event will be $\Pr(\textsc{Event}\text{ is true})$.

In the following, we study how to compute the prior probabilities of Presence and Pattern events. For simplicity, the events in the rest of the paper are defined in consecutive time and use *start* and *end* to denote the start time point and end time point of the user-specified spatiotemporal event. We assume $\mathbb{S}$ to be $\{s_1, s_2, s_3\}$ in the following examples.

3.2.1 Presence Events

Example 3.1. Let us consider the same Presence event defined in Example 2.1, It is defined as an event passing through s_1 or s_2 during $t = 3$ or $t = 4$, i.e., $\mathbf{s} = [1, 1, 0]^\intercal$, $start = 3, end = 4$. The transition matrix $\mathbf{M}$ is given below:

$$\mathbf{M} = \begin{bmatrix} 0.1 & 0.2 & 0.7 \\ 0.4 & 0.1 & 0.5 \\ 0 & 0.1 & 0.9 \end{bmatrix}. \tag{2}$$

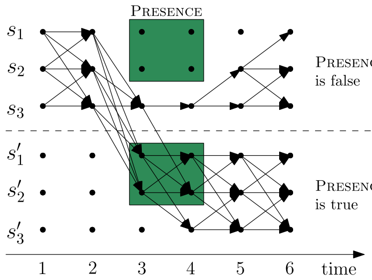


Fig. 5. New Markov transitions: all transitions going to the PRESENCE region will be re-directed to the virtual worlds.

Then Fig. 5 shows the new transitions in the two worlds, the top world and the bottom world separated by the dashed line in Fig. 5, corresponding to the two possible worlds where the presence event is false or true respectively. From time 1 to 2, a normal transition can be made. At timestamp 2, all the transitions going to the states s_1 and s_2 will be re-directed to the new states s'_1 and s'_2, denoting the states when the PRESENCE happens. Other transitions that do not go to the region will be performed normally. Similarly at time 3, the transition from s_3 to s_2 will also go to the state s'_2 because the event is also true in this case. After time 4, the original Markov transitions come back to work again.

The intuition can be formalized as follows. First, the original probabilities in $\mathbb{R}^{1\times m}$ is extended to $\mathbb{R}^{1\times 2m}$. Thus the initial probability $\boldsymbol{\pi}$ becomes $[\boldsymbol{\pi}, \mathbf{0}]$. Second, the transition matrix $\mathbf{M}_t$ takes the form of four transition matrices between the two virtual worlds, i.e., the EVENT is true or false, in Eq. (3). Then the new transition matrix can be derived in Eqs. (4) and (5) for different time period where $\mathbf{M}$ is the original transition matrix and s^D is the diagonal of the region s of PRESENCE defined in Definition 2.2

$$\mathbf{M}_t = \begin{bmatrix} \text{false} \rightarrow \text{false} & \text{false} \rightarrow \text{true} \\ \text{true} \rightarrow \text{false} & \text{true} \rightarrow \text{true} \end{bmatrix} \text{on the event.} \tag{3}$$

$$\mathbf{M}_t = \begin{bmatrix} \mathbf{M} - \mathbf{M}\mathbf{s}^{\mathbf{D}} & \mathbf{M}\mathbf{s}^{\mathbf{D}} \\ \mathbf{0}^{\mathbf{D}} & \mathbf{M} \end{bmatrix}, start - 1 \leq t \leq end - 1. \tag{4}$$

$$\mathbf{M}_t = \begin{bmatrix} \mathbf{M} & \mathbf{0}^{\mathbf{D}} \\ \mathbf{0}^{\mathbf{D}} & \mathbf{M} \end{bmatrix}, t < start - 1 \ or \ t \geq end. \tag{5}$$

Eq. (4), designed to capture and maintain all the transitions going to the region of the PRESENCE, is the new transition matrix when entering (and inside) the event time. Eq. (5), designed to keep the original transitions in the two virtual worlds, is the new transition matrix when leaving (and before) the event time. Third, at the last time T, the probability of the PRESENCE will be the sum of all probabilities in the bottom world (where PRESENCE is true).

3.2.2 Pattern Events

For PATTERN events, the bottom world denoting the event is true only needs to preserve the transitions going to the defined regions of the PATTERN event. The following example shows the mechanism.

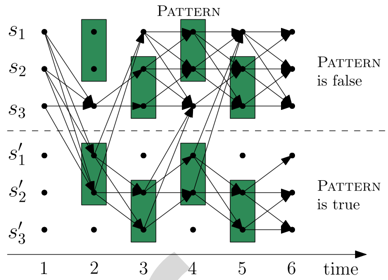


Fig. 6. New Markov transitions: at timestamp 1, all transitions going to the defined region will be re-directed to the bottom world; at timestamp $2 \sim 4$, only the transitions from the bottom world to the defined regions remain below.

Example 3.2. We study the PATTERN event as illustrated in Fig. 6. At time 1, the transitions entering s_1 and s_2 go to s'_1 and s'_2. From time 2 to 4, the transitions in the top world were performed normally. However, the transitions from the bottom world go back to the top world if the destinations are not in the defined regions. At time 5, the original Markov transitions come back to work again.

From above example, the transition matrices for PATTERN differ from the ones for PRESENCE during the event time from $start$ to $end - 1$ (i.e., Eq. (7)). On the other hand, when it is outside the event, i.e., $t < start - 1 \ or \ t \geq end$, the transition matrices for PATTERN are the same as the ones for PRESENCE (i.e., the matrices in (8) and (5) are the same). Finally, when $t = start - 1$, the transition matrices for PATTERN is also as same as the ones for PRESENCE (i.e., the matrices in (6) and (4) are identical)

$$\mathbf{M}_t = \begin{bmatrix} \mathbf{M} - \mathbf{M}\mathbf{s}^{\mathbf{D}} & \mathbf{M}\mathbf{s}^{\mathbf{D}} \\ \mathbf{0}^{\mathbf{D}} & \mathbf{M} \end{bmatrix}, t = start - 1. \tag{6}$$

$$\mathbf{M}_t = \begin{bmatrix} \mathbf{M} & \mathbf{0}^{\mathbf{D}} \\ \mathbf{M} - \mathbf{M}\mathbf{s}_t^{\mathbf{D}} & \mathbf{M}\mathbf{s}_t^{\mathbf{D}} \end{bmatrix}, start \leq t \leq end - 1. \tag{7}$$

$$\mathbf{M}_t = \begin{bmatrix} \mathbf{M} & \mathbf{0}^{\mathbf{D}} \\ \mathbf{0}^{\mathbf{D}} & \mathbf{M} \end{bmatrix}, t < start - 1 \ or \ t \geq end. \tag{8}$$

In summary, the prior probability of any EVENT can be derived as the sum of probabilities in the world where the EVENT is true. Lemma 3.1 shows the formal computation.

Lemma 3.1. *For initial probability $\boldsymbol{\pi} \in \mathbb{R}^{1\times m}$, the prior probability of an* EVENT *of* PRESENCE *and* PATTERN *is*

$$\Pr(\text{EVENT}) = [\boldsymbol{\pi}, \mathbf{0}] \prod_{i=1}^{end-1} \mathbf{M}_i [\mathbf{0}, \mathbf{1}]^{\intercal}, \tag{9}$$

where $\mathbf{M}_i$ is computed by Eqs. (4), (5), (6), (7), (8).

If the Markov model is time-varying, i.e., when the transition matrices $\mathbf{M}$ at different t are not identical, the only extra effort is to re-compute Eqs. (4)$\sim$(8) using the corresponding transition matrix $\mathbf{M}$ at t.

3.3 Computing Joint Probability of an Event

The calculation of a spatiotemporal event and a sequence of observed locations, i.e, $\Pr(o_1, o_2, \ldots, o_T, \text{EVENT})$ is a little more complicated than previous sections since it depends on not only the initial probabilities but also the emission matrix of the LPPM. Similarly, we use two-possible-world method to avoid enumerating all possible cases of an event. We utilize forward-backward algorithm[16] to estimate the probability of the true state (true location) at timestamp t given all observations $\Pr(l_t|o_1, o_2, \ldots, o_T)$. It first calculates a forward probability $\alpha_t^k = Pr(l_t = s_k, o_1, o_2, \ldots, o_t)$ iteratively, i.e.,

$$\alpha_t^k = Pr(o_t|l_t = s_k) \sum_i \alpha_{t-1}^i Pr(l_t = s_k|l_{t-1} = s_i). \tag{10}$$

Then, a backward probability $\beta_t^k = \Pr(o_{t+1}, o_{t+2}, \ldots, o_T|l_t = s_k)$ can also be derived by

$$\beta_t^k = \sum_i Pr(l_{t+1} = s_i|l_t = s_k) Pr(o_{t+1}|l_{t+1} = s_i) \beta_{t+1}^i. \tag{11}$$

By initializing $\beta_T^k = 1$ for all k, we can obtain the estimation of l_t as follows:

$$Pr(l_t = s_k|o_1, o_2, \ldots, o_T) = \frac{\alpha_t^k \beta_t^k}{\sum_i \alpha_t^i \beta_t^i}. \tag{12}$$

Intuition. The intuition of our solution is to use the forward-backward algorithm in the two virtual worlds where the EVENT is true and false. This is feasible because the emission probability, which determines the probabilities of the observations, is independent from any EVENTS. Hence in our computation the forward probability and backward probability are $\Pr(\text{EVENT}, o_1, o_2, \ldots, o_t)$ for $t \leq end$ and $\Pr(o_{end+1}, o_{end+2}, \ldots, o_t|\text{EVENT})$ for $t > end$ respectively. By combining them together, we can obtain the posterior probability of the EVENT. Note that at any timestamp $t \leq end$, we do not see the future ($t > end$) observations. Thus the posterior probability only counts to the current timestamp t.

Before and During the Event. In the forward algorithm, the probability $\alpha_t^k = \Pr(l_t = s_k, o_1, o_2, \ldots, o_t)$ is derived at timestamp t. We represent α_t^k in the vector form $\boldsymbol{\alpha}_t = [\alpha_1^1, \alpha_t^2, \ldots, \alpha_t^m]$. Then it can be derived as $\boldsymbol{\alpha}_t = (\boldsymbol{\alpha}_{t-1}\mathbf{M}_{t-1}) \circ \tilde{\mathbf{p}}_{o_t} = \boldsymbol{\alpha}_{t-1}\mathbf{M}_{t-1}\tilde{\mathbf{p}}_{o_t}^{\mathbf{D}}$. Without any further observations, the joint probability can be derived from Lemma 3.1. The result is shown in Lemma 3.2.

Lemma 3.2. *Given an initial probability $\boldsymbol{\pi}$, the joint probability of an* EVENT *of* PRESENCE *or* PATTERN *and observations $o_1, o_2, \ldots, o_t$ at any timestamp $t \leq end$ is*

$$\Pr(\text{EVENT}, o_1, o_2, \ldots, o_t) = [\boldsymbol{\pi}, \mathbf{0}] \left(\tilde{\mathbf{p}}_{o_1}^{\mathbf{D}} \prod_{i=2}^{t} (\mathbf{M}_{i-1}\tilde{\mathbf{p}}_{o_i}^{\mathbf{D}}) \prod_{i=t}^{end-1} \mathbf{M}_i [\mathbf{0}, \mathbf{1}]^{\intercal} \right). \tag{13}$$

After the Event. In the backward algorithm, $\beta_t^k = \Pr(o_{t+1}, o_{t+2}, \ldots, o_T|l_t = s_k)$. We represent it in the vector form $\boldsymbol{\beta}_t = [\beta_t^1, \beta_t^2, \ldots, \beta_t^m]$. Then it can be derived as $\boldsymbol{\beta}_t = (\boldsymbol{\beta}_{t+1} \circ \tilde{\mathbf{p}}_{o_{t+1}})\mathbf{M}_t = \boldsymbol{\beta}_{t+1}\tilde{\mathbf{p}}_{o_{t+1}}^{\mathbf{D}}\mathbf{M}_t$ for any $t > end$. Similarly, we have Lemma 3.3 for joint probability.

Lemma 3.3. *Given an initial probability $\boldsymbol{\pi}$, the joint probability of an* EVENT *of* PRESENCE *or* PATTERN *and observations $o_1, o_2, \ldots, o_t$ at any timestamp $t > end$ is*

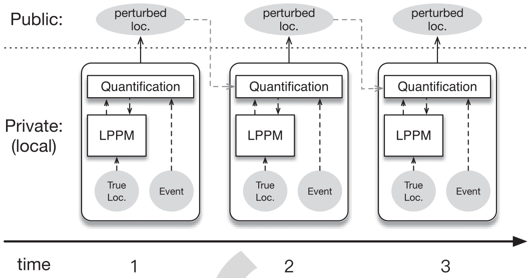


Fig. 7. PriSTE framework.

$$\Pr(\text{EVENT}, o_1, o_2, \ldots, o_t) = [\boldsymbol{\pi}, \mathbf{0}] \left(\tilde{\mathbf{p}}_{o_1}^{\mathbf{D}} \prod_{i=2}^{end} (\mathbf{M}_{i-1}\tilde{\mathbf{p}}_{o_i}^{\mathbf{D}}) \right) \left([\mathbf{1}, \mathbf{1}] \prod_{i=t-1}^{end} (\tilde{\mathbf{p}}_{o_{i+1}}^{\mathbf{D}} \mathbf{M}_i) \circ [\mathbf{0}, \mathbf{1}] \right)^{\intercal}. \tag{14}$$

To summarize, now we can quantify the ratio $\Pr(o_1, o_2, \ldots, o_T|\text{EVENT}) = \frac{Pr(o_1, o_2, \ldots, o_T, \text{EVENT})}{Pr(\text{EVENT})}$ for spatiotemporal event privacy using Lemma 3.1 to compute $\Pr(\text{EVENT})$ and Lemmas 3.2, 3.3 to compute $\Pr(o_1, o_2, \ldots, o_T, \text{EVENT})$. We note that our approach of computing the joint probability of an event is able to deal with different emission matrices at each t. Since $\tilde{\mathbf{p}}_{o_t}$ is a vector of emission probabilities given the observation o_t, i.e, a column in the emission matrix, and $\tilde{\mathbf{p}}_{o_t}^{\mathbf{D}}$ is a diagonal matrix whose diagonal elements are $\tilde{\mathbf{p}}_{o_t}$, we only need to obtain $\tilde{\mathbf{p}}_{o_t}$ and $\tilde{\mathbf{p}}_{o_t}^{\mathbf{D}}$ from the corresponding emission matrix at t, and then use such $\tilde{\mathbf{p}}_{o_t}^{\mathbf{D}}$ in Eqs. (13) and (14).

4 PRISTE FRAMEWORK

In previous section, we designed methods for quantifying ϵ-spatiotemporal event privacy provided by an LPPM w.r.t. a specified initial probability, which means that the privacy loss may not be bounded within ϵ if an attacker has a different initial probability.

In this section, we first design the Private Spatio-Temporal Event (PriSTE) framework and then solve the above problem by checking if ϵ-spatiotemporal event privacy for any initial probabilities. Finally, we demonstrate two case studies that instantiate the framework based different location privacy metrics for protecting spatiotemporal event privacy.

4.1 PriSTE

Based on the quantification techniques that we developed in previous sections, we propose a framework that converts a location privacy protection mechanism into one protecting spatiotemporal event privacy. The PriSTE framework is illustrated in Fig. 7 and described in Algorithm 1.

The major components are *Quantification* and a given LPPM. Their interactions are described as follows. First, the LPPM generates a perturbed location from the true location (Line 2 in Algorithm 1) and pass it to the *Quantification* component. By Theorem 4.1, the *Quantification* component (Line 3) checks whether this perturbed location satisfies the ratio in Eq. (1) (i.e., ϵ-spatiotemporal event privacy), under a sequence of previous observations and user-specified spatiotemporal events. If not, we need to calibrate the emission matrix to ensure that it satisfies ϵ-spatiotemporal event privacy. The strategy of emission matrix calibration is

LPPM-dependent. In the next section, we demonstrate case studies of Geo-Indistinguishability[8] and δ-location set privacy [11], which are the state-of-the-art location privacy metrics.

Algorithm 1. PriSTE Framework

Require: true location, ϵ, LPPM, $\mathbf{M}$, EVENTS
for t in $\{1, 2, \ldots, T\}$ **do**
  generate o_t with LPPM w.r.t. the true location;
  while ϵ-Spatiotemporal Event Privacy not hold **do**
    calibrate LPPM and generate o_t;
  end while
  release o_t;
end for

4.2 Privacy Checking With Arbitrary Initial Probability

According to the quantification in Section 3, we can calculate $\frac{\Pr(o_1,o_2,\ldots,o_T|\text{EVENT})}{\Pr(o_1,o_2,\ldots,o_T|\neg\text{EVENT})}$ given $o_1, o_2, \ldots, o_T$ and a given initial probability $\boldsymbol{\pi}$. In this section, we show how to make sure the ratio is bounded given arbitrary initial probability.

Our idea to is taking $\boldsymbol{\pi}$ as a variable and solving the maximization problem of $\frac{\Pr(o_1,o_2,\ldots,o_T|\text{EVENT})}{\Pr(o_1,o_2,\ldots,o_T|\neg\text{EVENT})} - e^{\epsilon}$. We want to make sure the maximum value is always less than or equal to 0, i.e., the user enjoys plausible deniability for her specified spatiotemporal event.

The following theorem shows the conditions related to $\boldsymbol{\pi}$ that satisfies ϵ-spatiotemporal event privacy. We will formulate it as an optimization problem.

Theorem 4.1. *For an* EVENT *of* PRESENCE *or* PATTERN *and an arbitrary initial probability $\boldsymbol{\pi}$, ϵ-spatiotemporal event privacy is satisfied at any timestamp t, i.e., $\frac{\Pr(o_1,o_2,\ldots,o_T|\text{EVENT})}{\Pr(o_1,o_2,\ldots,o_T|\neg\text{EVENT})} \leq e^{\epsilon}$, if the observation o_t is released based on the following two conditions*

$$\boldsymbol{\pi}\big([\mathbf{1}^{\mathrm{D}}, \mathbf{0}^{D}]((e^{\epsilon}-1)\boldsymbol{a}^{\intercal}\boldsymbol{b} - e^{\epsilon}\boldsymbol{a}^{\intercal}\boldsymbol{c})[\mathbf{1}^{D}, \mathbf{0}^{D}]^{\intercal}\big)\boldsymbol{\pi}^{\intercal} + \boldsymbol{\pi}[\mathbf{1}^{D}, \mathbf{0}^{D}](\boldsymbol{b}^{\intercal}) \leq 0 \tag{15}$$

$$\boldsymbol{\pi}\big([\mathbf{1}^{D}, \mathbf{0}^{D}]((e^{\epsilon}-1)\boldsymbol{a}^{\intercal}\boldsymbol{b} + \boldsymbol{a}^{\intercal}\boldsymbol{c})[\mathbf{1}^{D}, \mathbf{0}^{D}]^{\intercal}\big)\boldsymbol{\pi}^{\intercal} - \boldsymbol{\pi}[\mathbf{1}^{D}, \mathbf{0}^{D}](e^{\epsilon}\boldsymbol{b}^{\intercal}) \leq 0, \tag{16}$$

where

$$\boldsymbol{a} = \prod_{i=1}^{end-1} \boldsymbol{M}_i[\mathbf{0}, \mathbf{1}]^{\intercal}. \tag{17}$$

For $t \leq end$

$$\mathbf{b}^{\intercal} = \tilde{\mathbf{p}}_{o_1}^{\mathbf{D}} \prod_{i=2}^{t}(\mathbf{M}_{i-1}\tilde{\mathbf{p}}_{o_i}^{\mathbf{D}}) \prod_{i=t}^{end-1} \mathbf{M}_i[\mathbf{0}, \mathbf{1}]^{\intercal}\mathbf{c} = \tilde{\mathbf{p}}_{o_1}^{\mathbf{D}} \prod_{i=2}^{t}(\mathbf{M}_{i-1}\tilde{\mathbf{p}}_{o_i}^{\mathbf{D}})[\mathbf{1}, \mathbf{1}]^{\intercal}. \tag{18}$$

For $t > end$

$$\mathbf{b}^{\intercal} = \tilde{\mathbf{p}}_{o_1}^{\mathbf{D}} \prod_{i=2}^{end}(\mathbf{M}_{i-1}\tilde{\mathbf{p}}_{o_i}^{\mathbf{D}}) \left([\mathbf{1}, \mathbf{1}] \prod_{i=t-1}^{end} (\tilde{\mathbf{p}}_{o_{i+1}}^{\mathbf{D}}\mathbf{M}_i^{\intercal}) \circ [\mathbf{0}, \mathbf{1}]\right)^{\intercal} \tag{19}$$

$$\mathbf{c}^{\intercal} = \tilde{\mathbf{p}}_{o_1}^{\mathbf{D}} \prod_{i=2}^{end}(\mathbf{M}_{i-1}\tilde{\mathbf{p}}_{o_i}^{\mathbf{D}}) \left([\mathbf{1}, \mathbf{1}] \prod_{i=t-1}^{end} (\tilde{\mathbf{p}}_{o_{i+1}}^{\mathbf{D}}\mathbf{M}_i^{\intercal}) \circ [\mathbf{1}, \mathbf{1}]\right)^{\intercal}. \tag{20}$$

Quadratic Programming. To determine whether Eqs. (15) and (16) are true or not for arbitrary $\boldsymbol{\pi}$, we transform them to maximization problems: finding the maximum values of the left parts of Eqs. (15) and (16) under the constraints of $0 \leq p_i \leq 1$ where $p_i \in \boldsymbol{\pi}$. As long as one maximum value is larger than 0, we know that the LPPM (emission matrix) may not satisfy ϵ-spatiotemporal event privacy. The maximization are equivalent to quadratic programing problem since they can be rewritten in a form of $\boldsymbol{\pi}\mathbf{A}\boldsymbol{\pi}^{\intercal} = \frac{1}{2}\boldsymbol{\pi}(\mathbf{A} + \mathbf{A}^{\intercal})\boldsymbol{\pi}^{\intercal}$ where $\mathbf{A}$ is a matrix. We skip the computation details about solving such quadratic programing problem since many methods and tools have been proposed in literature. In the experiments, we use IBM CPLEX optimizer [17] as our computation engine.

4.3 Case Study 1: PriSTE With Geo-Indistinguishability

In this section, we instantiate PriSTE framework using α-Planar Laplace mechanism (α-PLM) which is designed for Geo-Indistinguishability[8]. We first show the computation details for quantifying ϵ-spatiotemporal event privacy by Theorem 4.1, and then design a greedy strategy for approximately achieving ϵ-spatiotemporal event privacy.

Algorithm Design. To implement the quantification component, we need to (1) compute the internal parameters **a**, **b** and **c** shown in Theorem 4.1, and (2) design a strategy to calibrate the emission matrix.

For the calibration strategy for Planar Laplace Mechanism (PLM) with a specified privacy budget α (which solely determines the shape of the output distribution), we exponentially decay its privacy budget because a smaller privacy budget implies stronger protection for location privacy and less information disclosure. In our algorithm, decay rate $\frac{1}{2}$ for the privacy budget in Line 19 of Algorithm 2 is a tunable parameter that provides a trade-off between efficiency and utility of the released locations. Setting a small value allows the algorithm converge faster, but at the cost of over-perturbing the location at each timestamp. In contrast, using a large value is less efficient but allows better utility.

A natural question is whether we can always find an α to release a perturbed location that satisfies Eq. (1). The answer is affirmative because α converges exponentially to 0. When $\alpha = 0$, it releases no useful information about the true location, i.e., uniformly returning a random location without using user's true position. It is easy to verify that the Eqs. (15) and (16) are always true in this situation.

Algorithm 2 shows the computation process. To boost the efficiency of our algorithm, we use intermediate matrices $\mathbf{A}$ and $\mathbf{B}$ to facilitate the computation of $\mathbf{b}$ and $\mathbf{c}$. At time 1, we initialize the variables as line $4 \sim 8$. At any time before and inside the EVENT, we compute the variables as line $10 \sim 11$. At any timestamps after the EVENT, the variables are derived as line $13 \sim 14$. Then we use quadratic programming methods to check Eqs. (15) and (16) to decide whether to release the o_t or not. If not, we generate a new o_t

with only half α, and repeat the above process again. Finally, we update the matrices $\mathbf{A}$ and $\mathbf{B}$ as line $21 \sim 25$. If $t = end$, in line 10, the product $\prod_{i=t}^{end-1} \mathbf{M}_i$ will be the identity matrix. In line 22, $\mathbf{M}_0$ is the identity matrix when $t = 1$. We note that for multiple EVENTS, Algorithm 2 can be executed multiple times for each EVENT.

Algorithm 2. PriSTE With Geo-Indistinguishability

Require: ϵ, EVENT, α-PLM, $\mathbf{M}_i, \forall i \in \{1, 2, \dots, T\}$
1: **for** t in $\{1, 2, \dots, T\}$ **do**
2: $\quad o_t \leftarrow \alpha$-PLM; $\triangleright$initial budget$= \alpha$
3: $\quad$ **if** $t == 1$ **then**
4: $\quad\quad \mathbf{a}^\intercal \leftarrow \prod_{i=1}^{end-1} \mathbf{M}_i[\mathbf{0}, \mathbf{1}]^\intercal$
5: $\quad\quad \mathbf{A} \leftarrow \mathbf{I}$ $\triangleright$identity matrix
6: $\quad\quad \mathbf{B} \leftarrow \mathbf{I}$
7: $\quad\quad \mathbf{b}^\intercal \leftarrow \tilde{\mathbf{p}}_{o_1}^{\mathbf{D}} \mathbf{a}^\intercal$
8: $\quad\quad \mathbf{c}^\intercal \leftarrow \tilde{\mathbf{p}}_{o_1}^\intercal$
9: $\quad$ **else if** $t <= end$ **then** $\triangleright$before and duringEVENT
10: $\quad\quad \mathbf{b}^\intercal \leftarrow \mathbf{A}\mathbf{M}_{t-1}\tilde{\mathbf{p}}_{o_t}^{\mathbf{D}} \prod_{i=t}^{end-1} \mathbf{M}_i[\mathbf{0}, \mathbf{1}]^\intercal$
11: $\quad\quad \mathbf{c}^\intercal \leftarrow \mathbf{A}\mathbf{M}_{t-1}\tilde{\mathbf{p}}_{o_t}^{\mathbf{D}}[\mathbf{1}, \mathbf{1}]^\intercal$
12: $\quad$ **else** $\triangleright$afterEVENT
13: $\quad\quad \mathbf{b}^\intercal \leftarrow \mathbf{A}\Big(([\mathbf{1}, \mathbf{1}]\tilde{\mathbf{p}}_{o_t}^{\mathbf{D}}\mathbf{M}_{t-1}^\intercal\mathbf{B}) \circ [\mathbf{0}, \mathbf{1}]\Big)^\intercal$
14: $\quad\quad \mathbf{c}^\intercal \leftarrow \mathbf{A}\Big(([\mathbf{1}, \mathbf{1}]\tilde{\mathbf{p}}_{o_t}^{\mathbf{D}}\mathbf{M}_{t-1}^\intercal\mathbf{B}) \circ [\mathbf{1}, \mathbf{1}]\Big)^\intercal$
15: $\quad$ **end if**
16: $\quad$ **if** Eqs. (15) and (16) hold **then** $\triangleright\epsilon$ is used here.
17: $\quad\quad$ release o_t; $\triangleright$okay to release o_t
18: $\quad$ **else**
19: $\quad\quad \alpha \leftarrow \frac{\alpha}{2}$, goto Line 2; $\triangleright$halve the budget
20: $\quad$ **end if**
21: $\quad$ **if** $t \leq end$ **then**
22: $\quad\quad \mathbf{A} \leftarrow \mathbf{A}\mathbf{M}_{t-1}\tilde{\mathbf{p}}_{o_t}^{\mathbf{D}}$ $\triangleright$update $\mathbf{A}$ by the real o_t
23: $\quad$ **else**
24: $\quad\quad \mathbf{B} \leftarrow \tilde{\mathbf{p}}_{o_t}^{\mathbf{D}}\mathbf{M}_{t-1}^\intercal\mathbf{B}$ $\triangleright$update $\mathbf{B}$ by the real o_t
25: $\quad$ **end if**
26: **end for**

Complexity. The internal parameters $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$ in Algorithm 2 need $O(mT)$ time to be evaluated. The major computational cost lies in the quadratic program for checking Eqs. (15) and (16). The complexity will be determined by the quadratic matrix $[\mathbf{1}^{\mathbf{D}}, \mathbf{0}^{\mathbf{D}}]\mathbf{a}^\intercal\mathbf{c}[\mathbf{1}^{\mathbf{D}}, \mathbf{0}^{\mathbf{D}}]^\intercal$. If it is positive definite, then the complexity is $O(m^3)$. Otherwise, with any negative eigenvalues, it will be NP-hard [18]. In our experiments, we use IBM CPLEX which can provide globally optimal results for quadratic program but may need a long computation time. We use a *conservative release* strategy to remedy this: we use a threshold to limit the computation time of quadratic program for checking Eqs. (15) and (16). It will not release a perturbed location unless the equations are true. Although it may lead to suboptimal solution in budget calibration, it always guarantees ϵ-spatiotemporal event privacy since every released locations satisfy Eqs. (15) and (16).

Privacy Analysis. PriSTE framework relies on a local model, i.e., the assumption that adversaries cannot obtain user's locally stored information as shown in Fig. 7. Although line 2 may be executed more than once at a timestamp t, Algorithm 2 still satisfies α'-Geo-Indistinguishability where α' is the final privacy budget used for releasing o_t because that is the only observation of attacker at time t. If we remove the assumption of local model, the above statements may not be true since attacker may observe the internal states of the algorithm (which is the privacy goal of *pan-privacy* [19]). Examples of internal states includes multiple o_t tested at t or the final α' used in the algorithm. Another assumption that may affect the privacy guarantee is the transition matrix $\mathbf{M}$, which we use it to model the correlations between locations and assume that it is given. It is an interesting future work to quantify the change of privacy loss in terms of ϵ-spatiotemporal event privacy if the ground truth of correlation is not the modeled one.

4.4 Case Study 2: PriSTE With δ-Location Set Privacy

To evaluate the effectiveness of PriSTE under different location privacy protection mechanisms, we also instantiate it using another privacy metrics *δ-location set privacy*[11], [20], which is proposed for obtaining better utility by taking advantage of temporal correlation between consecutive locations in user's trajectory. The key idea is that hiding the true location in any impossible locations (e.g., whose probabilities are close to 0) is a lost cause because the adversary already knows the user cannot be there. In other words, it restricts the output domain of the emission matrix to *δ-location set*, which is a set containing minimum number of locations that have prior probability sum no less than $1 - \delta$. A larger δ indicates weaker privacy guarantee.

The privacy metrics of α-Geo-Indistinguishability and δ-location set privacy are orthogonal because the former requires a specific "shape" of emission distribution and the latter restricts output domain of the emission distribution. In [11], Xiao and Xiong proposed a framework to achieve δ-location set privacy using a given LPPM. For ease of comparison, we use α-PLM as the underlying LPPM for δ-location set privacy.

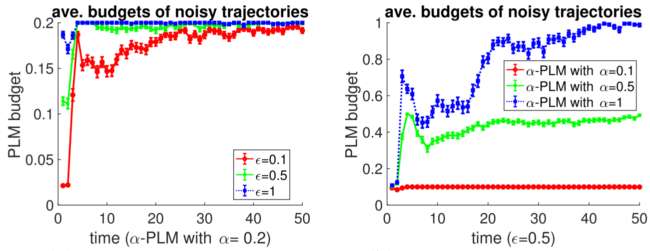
Algorithm 3. PriSTE With δ-Location Set Privacy

Require: ϵ, EVENT, α-PLM, $\mathbf{M}_i, \forall i \in \{1, 2, \dots, T\}$, π, δ, $\mathbf{M}$.
1: **for** t in $\{1, 2, \dots, T\}$ **do**
2: $\quad \mathbf{p}_t^- \leftarrow \mathbf{p}_{t-1}^+\mathbf{M}$; $\triangleright$Markov transition
3: $\quad$ Construct $\Delta\mathbf{X}_t$ $\triangleright\delta$-location set
4: $\quad o_t \leftarrow \alpha$-PLM within $\Delta\mathbf{X}_t$;
5: $\quad$ the same as Lines $3 \sim 15$ in Algorithm 2;
6: $\quad$ **if** Eqs. (15) and (16) hold **then** $\triangleright\epsilon$ is used here.
7: $\quad\quad$ release o_t; $\triangleright$okay to release o_t
8: $\quad\quad$ Derive posterior probability $\mathbf{p}_t^+$ by Eq. (21);
9: $\quad$ **else**
10: $\quad\quad \alpha \leftarrow \frac{\alpha}{2}$, goto Line 4; $\triangleright$halve the budget
11: $\quad$ **end if**
12: $\quad$ the same as Lines $21 \sim 25$ in Algorithm 2;
13: **end for**

In Line 2, when $t = 1$, we have $\mathbf{p}_0^+ = \pi$. In Line 8, according to [11], the posterior probability can be calculated by Eq. (21) where $\mathbf{p}_t^+[j]$ and $\mathbf{p}_t^-[i]$ are ith elements in the corresponding probability vectors

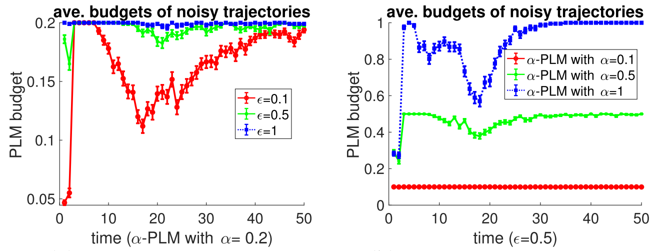
$$\mathbf{p}_t^+[i] = \Pr(l_t = s_i|o_t) = \frac{\Pr(o_t|l_t = s_i) * \mathbf{p}_t^-[i]}{\sum_j \Pr(o_t|l_t = s_j) * \mathbf{p}_t^-[j]}. \qquad (21)$$

Hence, we need the initial probability π in order to calculate δ-location set. In experiments, we set π to a uniform



(a) 0.2-PLM for different ϵ. (b) Different PLMs for $\epsilon = 0.5$.

Fig. 8. Presence($\mathcal{S} = [1:10], \mathcal{T} = [4:8]$) on synthetic data.



(a) 0.2-PLM for different ϵ. (b) Different PLMs for $\epsilon = 0.5$.

Fig. 9. Presence($\mathcal{S} = [1:10], \mathcal{T} = [16:20]$) on synthetic data.

distribution for the evaluation of δ-location set privacy. We note that PriSTE is agnostic to such initial probability since it guarantees spatiotemporal event privacy against adversaries with arbitrary knowledge about the initial probability.

5 Experimental Evaluation

In experiments, we verified that Algorithms 2 and 3 can adaptively calibrate the privacy budget of Planar Laplace Mechanism (PLM) at each timestamp for both location privacy and spatiotemporal event privacy. Especially, we highlight the following empirical findings.

- A stricter LPPM can satisfy a certain level of spatiotemporal event privacy *without* any change (i.e., no need of privacy budget calibration), whereas a more loose LPPM may need to reduce its privacy budget significantly for protecting the same event.
- For achieving the same level of ϵ-spatiotemporal event privacy using different LPPMs, a stricter LPPM is *not* always better in terms of data utility.
- If the user's transition matrix has a significant pattern, an LPPM may need a small privacy budge to achieve ϵ-spatiotemporal event privacy.

5.1 Experiment Settings and Metrics

Dataset. We used real-life and synthetic datasets in experiments. Geolife data [21] was collected from 182 users in a period of over three years. It recorded a wide range of users' outdoor movements, represented by a series of tuples containing latitude, longitude and timestamp. The user's entire trajectory is used to train the transition matrix M, e.g., with R package "markovchain".

We generated a synthetic trajectory and its transition probability matrix as follows. First, a map with $20 * 20$ cells is generated. Then, the transition probability from one cell to another is drawn from the two-dimensional Gaussian distribution with scale parameter σ based on the distance between the cells. Here, a smaller σ indicates that the user moves to the adjacent cells with a higher probability, i.e., the transition matrix has a more significant pattern. Finally, we produced trajectories with 50 timestamps using such transition matrix to simulate movement of a user.

Quadratic Programming. We use the IBM CPLEX optimizer 12.7.1 [17] to find the globally optimal solution for the quadratic programming in Algorithm 2. We adopt a strategy of *conservative release* as mentioned previously and limit the computation time for each optimization to 1 second.

Events. We investigate Presence and Pattern events, which are represented by two parameters $\mathcal{S}$ and $\mathcal{T}$. For example, Presence($\mathcal{S} = \{1:10\}, \mathcal{T} = [4:8]$) is Presence event denoting the user appears in the region of $\{s_1, s_2, \ldots, s_{10}\}$ during timestamps $\{4, 5, 6, 7, 8\}$.

Utility Metrics. We use two metrics to evaluate data utility.

- Privacy budget α used in PLM, including α at each timestamp (see Section 5.2) and the average α during the whole time period (see Section 5.3). A higher privacy budget indicates higher utility.
- The euclidean distance between the perturbed locations and the true locations. A smaller euclidean distance indicates higher utility.

We run our algorithm 100 times and aggregate the results to calculate average privacy budget and euclidean distance.

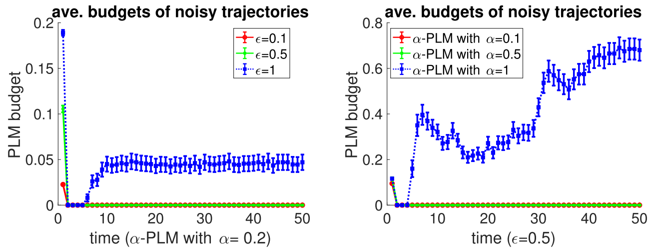
5.2 Utility at Each Timestamp

In this section, we show the utility (average privacy budget over 100 runs) at each timestamp for protecting Presence($\mathcal{S} = [1:10], \mathcal{T} = [4:8]$) and Presence($\mathcal{S} = [1:10], \mathcal{T} = [16:20]$). Due to the space limitation, we only show the results on synthetic data. We could draw the same conclusions from the results on Geolife data.

PriSTE with Geo-Indistinguishability . In Fig. 8a, it turns out that, 0.2-PLM satisfies 1-spatiotemporal event privacy with only slight privacy budget reduction, and satisfies 0.5-spatiotemporal event privacy with few budget reduction, but need to reduce more privacy budgets (to be stricter) in order to achieve 0.1-spatiotemporal event privacy. Similar results can be observed in Figs. 8b and 9. We also observe that the standard deviation is larger for weaker LPPMs since these privacy budgets need to be frequently calibrated. Hence, we can conclude that a stricter PLM for location privacy can protect spatiotemporal event without much calibration, but a more loose PLM may need to reduce its privacy budget significantly for ϵ-c.

In other words, in order to achieve a certain level of spatiotemporal event privacy, we need to sacrifice extra utility of an LPPM if the protection of the LPPM is weak (i.e., using a large privacy budget); as shown in the red lines in Figs. 8a and 9a, the LPPM, i.e., 0.2-PLM needs to reduce its budgets significantly for satisfying 0.1-spatiotemporal event privacy. On the other hand, we may not need to sacrifice utility of an LPPM to achieve the same level of spatiotemporal event privacy when the LPPM protection is strong (i.e., using a small privacy budget); as shown in the red lines in Figs. 8a and 9a, the LPPM, i.e., 0.2-PLM does not reduce its budgets significantly for 1-spatiotemporal event privacy.

We can see that the red line in Fig. 9a and the blue line in Fig. 9b during timestamps 10-30 occur larger standard deviation. Essentially, this is due to the fact that these PLMs are



(a) 0.2-PLM for different ϵ. (b) Different PLMs for $\epsilon = 0.5$.

Fig. 10. Protecting two events PRESENCE($\mathcal{S} = [1:10], \mathcal{T} = [4:8]$) and PRESENCE($\mathcal{S} = [1:10], \mathcal{T} = [16:20]$) on synthetic data.

reducing budgets significantly. The standard deviation is higher because the perturbation of the LPPM is enhanced with a smaller α.

Comparing Fig. 8 with Fig. 9, where the events are defined on time periods 4~8 and 16~20 respectively, we can see that privacy budgets tend to be reduced during the defined time periods. This indicates that the final α used by PLM at each timestamp may disclose the definition of spatiotemporal event as we discussed in Section 4.3. Hence, a local model is needed for PriSTE framework.

Protecting Multiple Events. Fig. 10 depicts the utilities when protecting two events sequentially using Algorithm 2. We can see that the utility is much worse than protecting each single event in Figs. 8 or 9 because the algorithm needs to simultaneously check if ϵ-spatiotemporal event privacy is satisfied for *both* events at each time. If no perturbed location satisfying the privacy requirement of both events simultaneously, the algorithm needs to halve the privacy budget until finding an appropriate output.

PriSTE with δ-Location Set Privacy. In Fig. 11, we show the utility of PriSTE with LPPMs that satisfies δ-Location Set Privacy (Algorithm 3). Comparing Fig. 11 with Fig. 8, although both of them are using 0.2-PLM, the essential difference between them is the privacy metric: the former satisfies δ-location set privacy and the latter satisfies Geo-Indistinguishability, i.e., 0.2-PLM in Fig. 11 has a constrained output domain. We can see that such 0.2-PLM in Fig. 11 has to reduce more privacy budgets to achieve ϵ-spatiotemporal event privacy. Intuitively, it is because the privacy metrics of δ-location set privacy implies a weaker privacy guarantee and its LPPM has to be stricter (using a smaller privacy budget) for protecting the event.

5.3 Utility Over Timestamps

In this section, we demonstrate the utility against different factors on the Geolife data and synthetic data. Figs. 12 and 13 are for protecting PRESENCE event. Due to the space limitation,

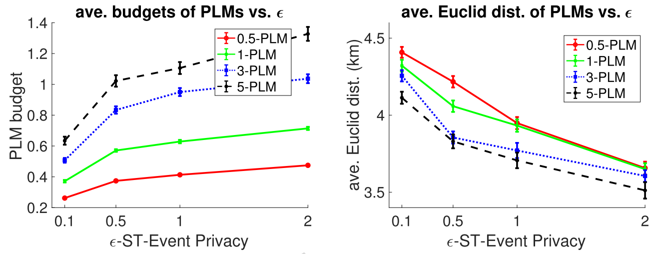


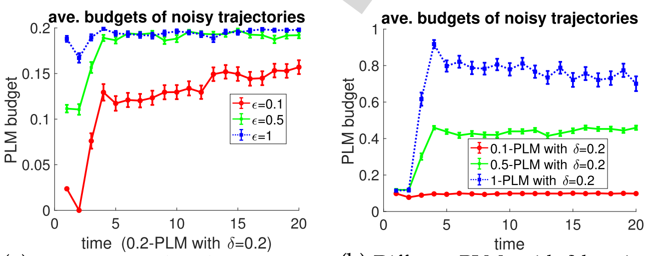
Fig. 12. PRESENCE($\mathcal{S} = [1:10], \mathcal{T} = [4:8]$) on Geolife data.

the results of protecting PATTERN event are included in Appendices, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TKDE.2019.2963312. Different from the utility in previous section which is averaged at each time, this section displays the utility that is further averaged over timestamps. Hence, in the left parts of Figs. 12 and 13 (ave. budget), the steeper lines indicate the budget may be reduced heavily at some timestamps. Generally, the utility increases with a larger ϵ in Figs. 12 and 13.

Utility versus α- Geo-Indistinguishability. In Fig. 12, we can see that a larger α-PLM needs to be calibrated heavily (i.e., steeper) for a small ϵ. Interestingly, PLMs with larger average budgets (in the left figures) may not necessarily have better utility in terms of euclidean distance. For example, at $\epsilon = 0.5$, the euclidean distance of 5-PLM and 3-PLM are very close; at $\epsilon = 1$ or 2, 0.5-PLM and 1-PLM appear to have almost the same euclidean distance. The reason is that PLMs that have larger *average* budgets may have extremely small budgets at some timestamps, which results in the higher *average* euclidean distance over timestamps.

Utility versus δ-Location Set Privacy. In Fig. 13, we can see that a PLM with a larger δ tends to have a smaller average budget. It is because the PLM with a larger δ indicates a weaker privacy metrics. Hence, the PLM needs to be stricter (i.e., a small budget) to achieve spatiotemporal event privacy. However, such PLM may have a better utility in terms of euclidean distance as shown in right figure of Fig. 13. The reason is that δ-location set privacy with a larger δ restricts the output domain significantly, which makes perturbed location close to the true location with a high probability. The results are in line with the main purpose of δ-location set privacy: to achieve a better privacy-utility trade-off.

Utility versus Transition Matrices. We compare the utility against transition matrices that have different strength of mobility patterns. As we explained previously, a smaller σ indicates a more significant mobility pattern. Fig. 14 shows that, for the same LPPM, it is hard to protect a spatiotemporal



(a) 0.2-PLM with δ-location set privacy for different ϵ. (b) Different PLMs with δ-location set privacy for $\epsilon = 0.5$.

Fig. 11. PRESENCE($\mathcal{S} = [1:10], \mathcal{T} = [4:8]$) on synthetic data.

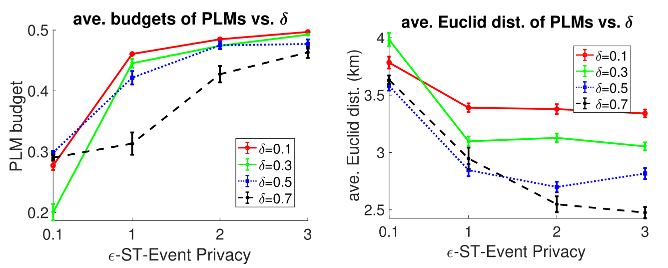


Fig. 13. PRESENCE($\mathcal{S} = [1:10], \mathcal{T} = [4:8]$) on Geolife data (0.5-PLM with δ-location set privacy).

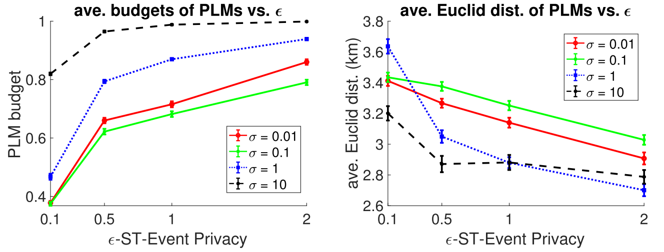


Fig. 14. PRESENCE($\mathcal{S} = [1:10], \mathcal{T} = [4:8]$) on synthetic data (1-PLM with Geo-Indistinguishability).

event if user's mobility pattern is significant, i.e., the LPPM needs to be very strict by using a small privacy budget. We also observe that there is no best LPPM for all ϵ-spatiotemporal event privacy in terms of euclidean distance.

5.4 Runtime

We name the size of $\mathcal{T}$ and the size of $\mathcal{S}$ as *event length* and *event width*, respectively. We also report the performance evaluation on *conservative release* described in Section 4.3.

Runtime versus Event Length. We fix the event width as 5 and test 100 random events with length ranging from 5 to 15. Fig. 15 shows that the average runtime of the baseline is exponential to event length and the runtime of our method is linear to the event length.

Runtime versus Event Width. We fix the event length as 5 and test 100 random events with width ranging from 5 to 15. Fig. 15 shows that the average runtime of the baseline is exponential to the event width, while our method is polynomial to the event width, which is in line with the complexity of $O(m^3)$.

Runtime versus Conservative Release. In Line 16 of Algorithm 2, we set a threshold runtime in solving the quadratic program. We do not release the perturbed location unless we are sure that Eqs. (15) and (16) are true. The threshold is a trade-off between runtime and utility as shown in Table 2 among 100 runs. We note that each runtime in Table 2 includes the whole process of Algorithm 2. In our implementation, we set the threshold to 1 second. We can see as the threshold increases, the number of conservative releases decreases, which results in increasing runtime. On the other hand, the calibrated privacy budgets increase as the threshold increases. This verifies the tradeoff between runtime and utility that can be achieved by the strategy of conservative release.

6 RELATED WORKS

6.1 Location Privacy Preserving Mechanisms

Existing works on location privacy can be roughly classified into two categories. The first type is the aggregated setting

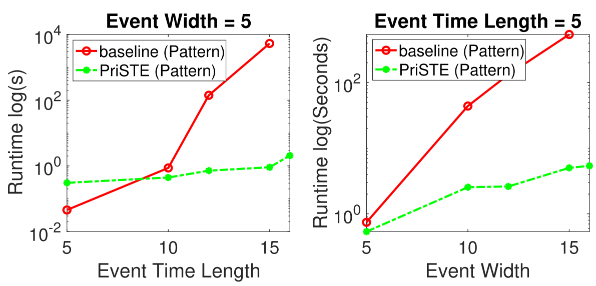


Fig. 15. Runtime evaluation.

TABLE 2
Runtime versus Threshold

threshold (s)	ave. total runtime (s)	# of Conservative Release	ave. privacy budget	ave. euclidean dist. (km)
0.01	1.1	**33**	0.16	2.22
0.1	2.6	30	0.23	1.51
1	5.9	21	0.22	1.52
2	10.4	12	0.29	**0.93**
5	19.5	8	0.27	1.41
none	**52.5**	0	**0.31**	0.97

[22], [23], [24], [25], [26], [27], where the goal is to protect the existence of a user's trajectory or a user's location when releasing aggregate location statistics of a dataset that consists of location sequences of a population of users. For example, DPT [24] used differential privacy techniques to synthesize a set of user trajectories based on statistical information that guarantees differential privacy. This is different from our problem setting of an individual user in location-based applications. The second type is the individual setting, which is also our setting, to protect the user's location when interacting with some location-based services. The LPPMs [8], [11], [28], [29], [30], [31] generally use some obfuscation methods, like spatial cloaking, cell merging, location precision reduction or dummy cells, to manipulate the probability distribution of users' locations. As differential privacy becomes a standard for privacy protection, [8] proposed a Geo-Indistinguishability notion based on differential privacy and a planar Laplace mechanism to achieve it. Xiao *et al.* [11], [20] studied how to protect location privacy under temporal correlations with an optimal differentially private mechanism. Rodriguez-Carrion *et al.* [32] also studied the effect of temporal dependencies on entropy-based location privacy metrics. They proposed a new privacy metrics *entropy rate* and perturbative mechanisms based on it, which can be an alternative LPPM in our framework for protecting spatiotemporal event privacy. Several studies [33], [34], [35] tried to achieve an optimal trade-off between the utility of applications and the privacy guarantee in the LPPMs. Overall, above works all focused on the mechanisms of location privacy, which can be used in our framework as given LPPMs. Whereas we study a new problem of spatiotemporal event privacy.

6.2 Inferences on Location

Various inference attacks can be carried out based on location information and external information such as moving patterns. In the aggregated setting, recent works have studied location or trajectory recovery attacks from aggregated location data[6], [36] or proximity query results from location data [4]. We mainly discuss the individual setting that is closely related to our work. Studies [33], [37], [38] investigated the question of how to formally quantify the privacy of existing LPPMs, given an adversary who can model users' mobility using a Markov process learned from population, which is commonly used for modeling user mobility pattern. Liao *et al.* [39] used a hierarchical Markov model to learn and infer a user's trajectory based on the places and temporal patterns they visited. Qiao *et al.* [40] used the Continuous Time Bayesian Networks to predict uncertain trajectories of moving objects. Li *et al.* [41] used frequent mining approach to find moving objects that move within arbitrary shape of clusters for certain timestamps that are possibly nonconsecutive.

7 Conclusion and Future Work

In this paper, we investigate a new type of pivacy goal called spatiotemporal event. We formally define spatiotemporal events and design a privacy metrics extending the notion of differential privacy. We proposed PriSTE, a framework integrating an LPPM for protecting the spatiotemporal event privacy. An interesting direction is to find optimal way for achieving both location privacy and spatiotemporal event privacy. Another question is how we can design a generic mechanism for spatiotemporal event privacy.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 17H06099, 18H04093, 19K20269, Microsoft Research Asia (CORE16), US National Science Foundation (NSF) under Grant No. CNS-1618932 and the AFOSR DDDAS program under grant No. FA9550-121-0240. This paper is extended from [1]. Yang Cao and Yonghui Xiao contributed equally to this work.

References

[1] Y. Cao, Y. Xiao, L. Xiong, and L. Bai, "PriSTE: From location privacy to spatiotemporal event privacy," in *Proc. IEEE 35th Int. Conf. Data Eng.*, 2019, pp. 1606–1609.

[2] P. Golle and K. Partridge, "On the anonymity of Home/Work location pairs," in *Proc. Int. Conf. Pervasive Comput.*, 2009, pp. 390–397.

[3] R. Recabarren and B. Carbunar, "What does the crowd say about you? Evaluating aggregation-based location privacy," in *Proc. Privacy Enhancing Technol.*, 2017, pp. 156–176.

[4] G. Argyros, T. Petsios, S. Sivakorn, A. D. Keromytis, and J. Polakis, "Evaluating the privacy guarantees of location proximity services," *ACM Trans. Privacy Secur.*, vol. 19, no. 4, pp. 12:1–12:31, 2017.

[5] K. Chatzikokolakis, E. ElSalamouny, C. Palamidessi, and P. Anna, "Methods for location privacy: A comparative overview," *Found. Trends Privacy Secur.*, vol. 1, no. 4, pp. 199–257, 2017.

[6] B. Liu, W. Zhou, T. Zhu, L. Gao, and Y. Xiang, "Location privacy and its applications," *IEEE Access*, vol. 6, pp. 17 606–17 624, 2018.

[7] V. Primault, A. Boutet, S. B. Mokhtar, and L. Brunie, "The long road to computational location privacy: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2772–2793, Jul.–Sep. 2019.

[8] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2013, pp. 901–914.

[9] C. Dwork, "Differential privacy: A survey of results," in *Proc. Int. Conf. Theory Appl. Models Comput.*, 2008, pp. 1–19.

[10] K. Chatzikokolakis, C. Palamidessi, and M. Stronati, "A predictive differentially-private mechanism for mobility traces," in *Proc. Int. Symp. Privacy Enhancing Technol. Symp.*, 2014, pp. 21–41.

[11] Y. Xiao and L. Xiong, "Protecting locations with differential privacy under temporal correlations," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, 2015, pp. 1298–1309.

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

[13] 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 *Proc. Workshop Privacy Electron. Soc.*, 2014, pp. 73–82.

[14] Y. De Mulder, G. Danezis, L. Batina, and B. Preneel, "Identification via location-profiling in GSM networks," in *Proc. Workshop Privacy Electron. Soc.*, 2008, pp. 23–32.

[15] M. Asada, M. Yoshikawa, and Y. Cao, "When and where do you want to hide?" – Recommendation of location privacy preferences with local differential privacy," in *Proc. IFIP Annu. Conf. Data Appl. Secur. Privacy*, 2019, pp. 164–176.

[16] B. SchusterBöckler and A. Bateman, "An introduction to hidden Markov models," *Current Protocols Bioinf.*, vol. 18, no. 1, pp. A.3A.1–A.3A.9, 2007.

[17] IBM CPLEX optimizer, 2017, [Online]. Available: https://www.ibm.com/support/pages/downloading-ibm-ilog-cplex-optimization-studio-v1271

[18] P. M. Pardalos and S. A. Vavasis, "Quadratic programming with one negative eigenvalue is NP-hard," *J. Global Optim.*, vol. 1, no. 1, pp. 15–22, 1991.

[19] C. Dwork, M. Naor, T. Pitassi, G. N. Rothblum, and S. Yekhanin, "Pan-private streaming algorithms," in *Proc. Innovations Comput. Sci.*, 2010, pp. 66–80.

[20] Y. Xiao, L. Xiong, S. Zhang, and Y. Cao, "LocLok: Location cloaking with differential privacy via hidden Markov model," *Proc. VLDB Endowment*, vol. 10, no. 12, pp. 1901–1904, 2017.

[21] Y. Zheng, X. Xie, and W.-Y. Ma, "GeoLife: A collaborative social networking service among user, location and trajectory," *IEEE Data Eng. Bull.*, vol. 33, no. 2, pp. 32–39, Jun. 2010.

[22] K. Jiang, D. Shao, S. Bressan, T. Kister, and K.-L. Tan, "Publishing trajectories with differential privacy guarantees," in *Proc. 25th Int. Conf. Sci. Statist. Database Manage.*, 2013, pp. 12:1–12:12.

[23] Y. Cao and M. Yoshikawa, "Differentially private real-time data publishing over infinite trajectory streams," *IEICE Trans. Inf. Syst.*, vol. E99-D, no. 1, pp. 163–175, 2016.

[24] X. He, G. Cormode, A. Machanavajjhala, C. M. Procopiuc, and D. Srivastava, "DPT: Differentially private trajectory synthesis using hierarchical reference systems," *Proc. VLDB Endowment*, vol. 8, no. 11, pp. 1154–1165, 2015.

[25] Y. Cao, L. Xiong, M. Yoshikawa, Y. Xiao, and S. Zhang, "ConTPL: Controlling temporal privacy leakage in differentially private continuous data release," *Proc. VLDB Endowment*, vol. 11, no. 12, pp. 2090–2093, 2018.

[26] V. Bindschaedler and R. Shokri, "Synthesizing plausible privacy-preserving location traces," in *Proc. IEEE Symp. Secur. Privacy*, 2016, pp. 546–563.

[27] Y. Cao and M. Yoshikawa, "Differentially private real-time data release over infinite trajectory streams," in *Proc. 16th IEEE Int. Conf. Mobile Data Manage.*, 2015, pp. 68–73.

[28] G. Ghinita, M. L. Damiani, C. Silvestri, and E. Bertino, "Preventing velocity-based linkage attacks in location-aware applications," in *Proc. 17th ACM SIGSPATIAL Int. Conf. Advances Geographic Inf. Syst.*, 2009, pp. 246–255.

[29] C. A. Ardagna, M. Cremonini, S. D. C. di Vimercati, and P. Samarati, "An obfuscation-based approach for protecting location privacy," *IEEE Trans. Dependable Secure Comput.*, vol. 8, no. 1, pp. 13–27, Feb. 2011.

[30] R. H. Hwang, Y. L. Hsueh, and H. W. Chung, "A novel time-obfuscated algorithm for trajectory privacy protection," *IEEE Trans. Services Comput.*, vol. 7, no. 2, pp. 126–139, Apr.–Jun. 2014.

[31] S. Takagi, Y. Cao, Y. Asano, and M. Yoshikawa, "Geo-graph-indistinguishability: Protecting location privacy for LBS over road networks," in *Proc. IFIP Annu. Conf. Data Appl. Secur. Privacy*, 2019, pp. 143–163.

[32] A. Rodriguez-Carrion *et al.*, "Entropy-based privacy against profiling of user mobility," *Entropy*, vol. 17, no. 6, pp. 3913–3946, 2015.

[33] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec , and J.-P. Hubaux, "Quantifying location privacy," in *Proc. IEEE Symp. Secur. Privacy*, 2011, pp. 247–262.

[34] N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Optimal geo-indistinguishable mechanisms for location privacy," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2014, pp. 251–262.

[35] R. Shokri, "Privacy games: Optimal user-centric data obfuscation," *Proc. Privacy Enhancing Technol.*, vol. 2015, no. 2, pp. 299–315, 2015.

[36] F. Xu, Z. Tu, Y. Li, P. Zhang, X. Fu, and D. Jin, "Trajectory recovery from ash: User privacy is NOT preserved in aggregated mobility data," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 1241–1250.

[37] Y. Cao, M. Yoshikawa, Y. Xiao, and L. Xiong, "Quantifying differential privacy under temporal correlations," in *Proc. IEEE 33rd Int. Conf. Data Eng.*, 2017, pp. 821–832.

[38] Y. Cao, M. Yoshikawa, Y. Xiao, and L. Xiong, "Quantifying differential privacy in continuous data release under temporal correlations," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 7, pp. 1281–1295, Jul. 2019.

[39] L. Liao, D. Fox, and H. Kautz, "Learning and inferring transportation routines," in *Proc. Nat. Conf. Artif. Intell.*, 2004, pp. 348–353.

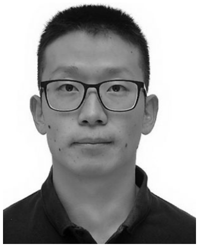
[40] S. Qiao *et al.*, "PutMode: Prediction of uncertain trajectories in moving objects databases," *Appl. Intell.*, vol. 33, no. 3, pp. 370–386, 2010.

[41] Z. Li, B. Ding, J. Han, and R. Kays, "Swarm: Mining relaxed temporal moving object clusters," *Proc. VLDB Endowment*, vol. 3, no. 1/2, pp. 723–734, 2010.

Yang Cao received the BS degree from the School of Software Engineering, Northwestern Polytechnical University, Xi'an, China, in 2008, and the MS and PhD degrees from the Graduate School of Informatics, Kyoto University, Kyoto, Japan, in 2014 and 2017, respectively. He is an assistant professor with the Department of Social Informatics, Kyoto University. He was a postdoctoral fellow with the Department of Math and Computer Science, Emory University. His research interests include privacy preserving data analysis, data trading, and distributed computing.

Yonghui Xiao received the three BS degrees from Xi'an Jiaotong University, Xi'an, China, in 2005, the MS degree from Tsinghua University, Beijing, China, in 2011 after spending two years working in industry, and the PhD degree from the Department of Math and Computer Science, Emory University, Atlanta, Georgia, in 2017. He is currently a senior software engineer with Google working on machine learning and cloud security.

Li Xiong received the BS degree from the University of Science and Technology of China, Hefei, China, the MS degree from Johns Hopkins University, Baltimore, Maryland, and the PhD degree from the Georgia Institute of Technology, Atlanta, Georgia, all in computer science. She is a Winship distinguished research professor of computer science (and biomedical informatics) with Emory University. She and her research group, Assured Information Management and Sharing (AIMS), conduct research that addresses both fundamental and applied questions at the interface of data privacy and security, spatiotemporal data management, and health informatics.



Liquan Bai received the BS degree in computer science from Tianjin University, Tianjin, China, and the two MS degrees in telecommunication and computer science from Northeastern University, Boston, Massachusetts and Emory University, Atlanta, Georgia, respectively. He had worked as a software engineer in industry for about 4.5 years. He is currently a research assistant with the Department of Computer Science, Emory University.

Masatoshi Yoshikawa received the BE, ME, and PhD degrees from the Department of Information Science, Kyoto University, Kyoto, Japan, in 1980, 1982 and 1985, respectively. Before joining Graduate School of Informatics, Kyoto University as a professor in 2006, he has been a faculty member of Kyoto Sangyo University, Nara Institute of Science and Technology, and Nagoya University. His general research interests include the area of databases. His current research interests include multiuser routing algorithms and services, theory and practice of privacy protection, and medical data mining. He is a member of the ACM and IPSJ.