Are Friends of My Friends Too Social? Limitations of Location Privacy in a Socially-Connected World

Boris Aronov
New York University
Brooklyn, NY, USA
boris.aronov@nyu.edu

Alon Efrat
University of Arizona
Tucson, AZ, USA
alon@email.arizona.edu

Ming Li
University of Arizona
Tucson, AZ, USA
lim@email.arizona.edu

Jie Gao
Stony Brook University
Stony Brook, NY, USA
jie.gao@stonybrook.edu

Joseph S. B. Mitchell
Stony Brook University
Stony Brook, NY, USA
joseph.mitchell@stonybrook.edu

Valentin Polishchuk
Linköping University
Norrköping, Sweden
valentin.polishchuk@liu.se

Boyang Wang
University of Cincinnati
Cincinnati, OH, USA
wang2ba@ucmail.uc.edu

Hanyu Quan
Xidian University
Xi'an, China
quanhanyu@gmail.com

Jiaxin Ding
Stony Brook University
Stony Brook, NY, USA
jiaxin.ding@stonybrook.edu

ABSTRACT

With the ubiquitous adoption of smartphones and mobile devices, it is now common practice for one's location to be sensed, collected and likely shared through social platforms. While such data can be helpful for many applications, users start to be aware of the privacy issue in handling location and trajectory data. Some users may voluntarily share their location information (e.g., for receiving location-based services, or for crowdsourcing systems), which may lead to information leaks about the whereabouts of other users, through the co-location of events when two users are at the same location at the same time and other side information, such as upper bounds on movement speed. It is therefore crucial to understand how much information one can derive about others' positions through the co-location of events and occasional GPS location leaks of some of the users. In this paper we formulate the problem of inferring locations of mobile agents, present theoretically-proven bounds on the amount of information that could be leaked in this manner, study their geometric nature, and give algorithms matching these bounds. We will show that even if a very weak set of assumptions is made on trajectories' patterns, and users are not obliged to follow any 'reasonable' patterns, one could obtain very accurate estimation of users' locations even if they opt not to share them. Furthermore, this information could be obtained using almost linear-time algorithms, suggesting the practicality of the method even for huge volumes of data.

CCS CONCEPTS

• **Security and privacy → Social aspects of security and privacy**; • **Theory of computation → Computational geometry**;

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Mobihoc '18, June 26–29, 2018, Los Angeles, CA, USA

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-5770-8/18/06…$15.00
https://doi.org/10.1145/3209582.3209611

ACM Reference Format:
Boris Aronov, Alon Efrat, Ming Li, Jie Gao, Joseph S. B. Mitchell, Valentin Polishchuk, Boyang Wang, Hanyu Quan, and Jiaxin Ding. 2018. Are Friends of My Friends Too Social? Limitations of Location Privacy in a Socially-Connected World. In *Mobihoc '18: The Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing, June 26–29, 2018, Los Angeles, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3209582.3209611

1 INTRODUCTION

With today's nearly universal use of smartphones, people rely on Location-Based Services (LBSes) on a daily basis. For instance, using Google Maps, a user can search for restaurants and landmarks in her proximity and can find the fastest route to a place from her current location. Moreover, in Location-Based Social Network (LBSN) applications, a user can locate her friends via geo-tagged posts on popular social networking platforms, such as Foursquare, Instagram, or Facebook. However, due to the sensitivity of location data, this use of LBSes also raises significant privacy concerns. For example, leakage of locations can lead to the exposure of sensitive personal information, such as home addresses, health conditions, sexual orientation, or political and religious beliefs [24]. A variety of location privacy metrics and protection mechanisms have been proposed in the context of LBSes, including perturbation-based approaches, such as the classical notions of k-anonymity [23] and spatial cloaking [14, 16, 22], and rigorous privacy-based definitions, such as differential privacy [13] and geo-indistinguishability [2], which can guarantee the indistinguishability of two locations to an adversary given the perturbed ones, provided arbitrary background information. Crypto-based approaches can also achieve privacy-preserving location-based queries against untrusted service providers with even stronger security guarantees, such as homomorphic encryption [21, 37], private information retrieval [15], searchable encryption [30, 31], etc.

While all existing techniques satisfy indistinguishability-based security/privacy requirements, it remains unclear what concrete location privacy guarantees they can provide in a real-world setting, especially against an adversary with certain background knowledge. In fact, several recent studies [3, 14, 33] have shown that the actual location privacy level of perturbation-based methods

can be quite limited when an attacker has *side information*. For example, one can use the maximum velocity of users to dramatically reduce the region in which a user could be, even if they cloak their locations [14]. In addition, other studies have shown that in proximity-based geo-query response systems, an adversary can carry out triangulation-type attacks to accurately pinpoint user locations, even if they are protected using crypto-based techniques [3, 19]. All of this work deals mainly with LBS applications from the perspective of single users; however, with the increasing use of social networking platforms, user–user interactions can provide richer side information that potentially leads to higher privacy leakage than expected, which is not well understood.

For a LBSN, our main observation is that even if a privacy-sensitive individual's location is well-protected by him/herself, it may still be revealed through other users' sharing behavior. In practice, users have different privacy settings and preferences in LBSNs. Some users who are less privacy-aware publish their locations directly on social media. For example, in Facebook, if Alice posts a photo taken with Bob, along with her current location (e.g., by geo-tagging him), then the location of Bob is immediately revealed. Similarly, suppose Charlie is a privacy-aware user on Foursquare, and he sometimes reports his actual locations, but hides the locations of special events (e.g., confidential meetings with his doctor); then, using the times of his published locations before and after the meeting, and some side information, such as whether he is driving or walking, his private meeting locations can be roughly inferred. Thus, we ask the fundamental question: how much information can be inferred about a user's location, given the co-location events with others and occasional GPS location leaks?

In this paper, we characterize the fundamental limit of location inference given limited available information in LBSNs, including sporadically published locations by users, and side information (such as meeting events and maximum speed of users). We propose a theoretical framework to bound the feasible region of users' unknown locations, by leveraging geometric constraints imposed by users' movement and meeting relationships.

Our results show that even with such a weak set of assumptions, it is still feasible to restrict users' unknown locations to small regions. Our attack model is generally applicable to a wide range of location privacy preserving techniques (such as cloaking, encryption, or changing privacy settings), while our approach does not make any assumptions on the statistical mobility patterns of users. The main contributions of this paper include:

- We characterize the *feasibility region* for the possible location of each agent, at each time. We prove that these regions are optimal, in the sense that they are as accurate as possible, given the available data: The agent could be anywhere inside the feasibility region, without violating any constraints; similarly, the agent could not be anywhere outside the feasibility region, without violating the constraints.
- We give exact and efficient polynomial-time algorithms for computing feasibility regions when distances are measured using the L_∞ norm (i.e., for computing the bounding boxes of the regions). If distances are measured using the Euclidean L_2 norm, we provide evidence that computing the exact feasibility regions is computationally impractical, but we propose efficient algorithms for ε-approximating them.
- We provide hardness results for computing the feasibility regions when obstacles are present, and then propose a constant-factor approximation algorithm.
- Using simulations on a synthetic dataset and a real GPS dataset, we evaluate the correctness and effectiveness of our algorithms. The results show that the location uncertainty decreases with an increasing number of known GPS locations and meeting events, which matches with intuition.

2 RELATED WORK

Attacks on Location Privacy. Early works showed that an attacker can easily learn a user's identity through anonymized GPS traces [18] (but accurate locations). User de-anonymization and location recovery attacks are often intertwined with each other, since with only a few locations of a user, one can uniquely recover that user's identity [10]. Many recent attacks [20, 27, 28] show that even if locations are protected (e.g., perturbed with noise or cloaking), location traces can still be de-anonymized. For instance, by exploiting prior knowledge of users' movement patterns (training a Markov transition matrix for each user), Shokri et al. [28] proposed optimal inference attacks against user's noisy locations, and used attacker's expected estimation error to quantify location privacy leakage under inference attacks. In a subsequent work [27], they consider the scenario when location data is sporadic, and quantified the location privacy leakage using Bayesian inference for Hidden Markov Processes, where an adversary knows users' movement patterns and geographical distribution. Recently, Murakami [20] improved this by leveraging the Viterbi algorithm and Forward Filtering Backward Sampling to build users' transition matrices when the size of training data is small and data itself is sporadic/sparse. In addition, Pyrgelis et al. [25] showed that individual location traces can be recovered even when only aggregated time-series location data is released.

All of the above works assumed that the attacker possesses the *prior knowledge of an individual user's mobility pattern*. In practice, this assumption is too strong. Although Ghinita *et al.* [14] presented a velocity-based attack to recover users' private locations when they are protected with spatial cloaking, it requires knowing users' precise speeds, and it is not effective with sporadic locations, since the attack relies on checking the intersection of two continuous cloaking regions. In contrast, our approach exploits the geometric properties to estimate unknown locations, based merely on *sporadic locations*, user-reported meeting events and maximum speed. While incorporating more side information such as prior location distribution or spatio-temporal correlation can improve the attack effectiveness, the uncertainty region derived in this paper can be regarded as an upper bound.

On the other hand, if an active adversary is considered, for example, in LBSN applications with geo-distance based retrieval, recent studies show that individual's hidden locations can be recovered using a small number of queries via triangulation [29] or space-partition based attacks [3, 19, 24]. Li et al. [19] first presented an attack against friend discovery services in LBSNs, by manipulating the input query locations and running proximity query as an oracle. Later, Polakis et al. [3, 24] improved such attack by formalizing it as a point search problem with a set of proximity queries, where they show a minimum logarithmic number of queries is sufficient. Further, Wu and Hu [33] proposed an entropy-minimization based

attack to recover the location of users' posts using a small number of adaptively chosen proximity queries and the corresponding query results, which works even when noise is added to the locations. However, all the above attacks target at a single location of a user, and they need the LBSNs to support interactive query-response. Our model is different, as we only consider passive attacks where locations can be leaked from information already available on an LBSN (e.g., posts/tagged photos by a friend of friend showing co-location).

Location Privacy Protection Techniques. There are three types of techniques to protect location privacy: anonymization-based, perturbation-based and crypto-based techniques. The first type requires a trusted server to publish a synthetic location dataset while preserving individual privacy [4, 5]. This is not applicable to our problem since the server may not be fully trusted and we are not concerned with data publishing. Perturbation-based methods do not assume a trusted server. Traditional *Location Obfuscation* methods such as location cloaking [16, 35] and k-anonymity [23], do not provide strong privacy guarantee and are prone to various attacks. *Differential Privacy (DP)* [13] was proposed that provides formal privacy guarantees that an adversary is not able to distinguish whether a particular individual participates in a published dataset, regardless of the amount of additional information available. Many recent techniques applied the principle of DP to location privacy protection, such as geo-indistinguishability [2] and δ-location set DP [34]. However, a common challenge or drawback with DP-based methods is that the data utility is often low if a reasonable privacy guarantee is needed, since the original data is perturbed according to the worst-case adversary assumption. Also, DP does not provide concrete guarantees in the face of attacks that aim to recover the original locations. Our proposed attack relaxes the assumption of the knowledge of the attacker, and provides a bound of the concrete location leakage under such attack.

On the other hand, crypto-based techniques offer strong privacy guarantees, but they are either not applicable to our setting, or are computationally inefficient over a large amount of data. For example, *Private Information Retrieval* (PIR) protocols [7, 15, 32] allow individual users to retrieve their nearest neighbors through an untrusted server, while the server learns nothing about the querying user's location. However, in this paper our goal is different (not to protect the privacy of users who share their locations, but to study the leakage of location privacy of all other users in an LBSN). Private proximity test protocols [17, 21, 36] allow two users to test their proximity without revealing their locations to each other. Also, recent searchable encryption techniques [30, 31] enable private geometric range queries over spatial data. While adapting crypto-based design to achieve some of the functions of an LBSN is possible, it may significantly affect the usability of an LBSN. Nevertheless, no matter what types of location privacy protection mechanism is adopted by users (be it perturbation or encryption), our model and approach are applicable.

3 PROBLEM STATEMENT

In this section, we first describe our problem along with the system and threat model, and then introduce main definitions and notation.

3.1 System and Threat Model

Consider a scenario in which a set of *users* (or *agents*) of a LBSN is moving in a two-dimensional domain; examples include shoppers in a mall, walkers in a rain forest, or vehicles on the road. In each of these scenarios, GPS signals are not always available/received so GPS coordinates are considered to be sporadic information. To use the LBSN, users may choose to report their location check-ins (including user ID, timestamps, and GPS locations) to a central server (e.g., Facebook). Depending on their own privacy settings, they may also choose not to report locations, or they may adopt other privacy enhancing techniques (such as cloaking or cryptographic tools) to protect their individual locations. In the case of cloaking, the server receives a region of uncertainty for the user's location. In the case of encryption, the location is considered to be unknown.

In addition, a history, or a *log* of events, such as meetings between pairs of agents, may be collected by the server and exchanged among the agents. For example, a user may tag his/her friends in a photo that documents their meeting, thereby indicating co-location; however, the meeting event may or may not have associated location data (e.g., if it took place in a rain forest, shopping mall, or other place where GPS is not available). Part of the location trace collection and event logs can be accessed by the general public, if some users put their privacy settings as public.

We consider an adversary (or *attacker*) to have access to the sporadically reported GPS locations of users, as well as the meeting event logs stored on the server. We also assume that this attacker has very limited side information, such as the maximum speed of agents, which can be obtained from knowing the coarse category of agents (pedestrians, bicycles or vehicles). Such an attacker could be the server itself, users of the social network service who are curious about other users' locations, or the general public. The server can be considered honest-but-curious (honestly executing the protocol but curious in knowing users' locations), or its stored data could be leaked out in the case of a data breach due to hacking. Note that we do not assume any other information (e.g., statistical user mobility patterns) is available to the adversary.

Our goal is to study how accurately the attacker can *infer* from the accumulated history of events, and any side information, the unknown or hidden locations of users who do not share their locations.

3.2 Definitions and Notation

We now give a more formal problem statement. There is a set, $\mathcal{A}$, of n agents; for simplicity of notation, we refer to agents by index (e.g. "agent i"), writing $\mathcal{A} = \{1, 2, \ldots, n\}$. We assume the agents move in a two-dimensional plane. GPS is able to provide location data for an agent only sporadically. When a pair of agents meet (e.g. when the distance between them allows bluetooth connections), they detect each other and register the meeting as an event. We assume that the attacker has access to logs (history) of all meetings among agents; the attacker will use this data to estimate where each meeting took place. Moreover, the attacker also knows an upper bound, $\bar{v}_i$, on the speed of each agent $i \in \mathcal{A}$. We distinguish between two types of events:

GPS events: A *GPS event* is a triple (i, τ, p), indicating that agent i is at location p at time τ.

Meeting events: A *meeting event* is a triple $\chi = (i, j, \tau)$ indicating that there was a meeting between agents i and j at time τ. Note that while the time, τ, of the meeting is specified, the location is not specified or required. In some cases, the location of a meeting event might known, in which case it will be separately specified via a GPS event associated

with one of the meeting agents at the time τ of the meeting. The attacker, however, may be able to infer locations or approximate locations of meeting events, as we describe below.

We let $\mathcal{E}_{GPS}$ denote the set of all GPS events and let $\mathcal{E}_{Meetings}$ denote the set of all meeting events.

Distance Constraints. The maximum speed, $\bar{v}_i$, of each agent i imposes constraints on the positions of agents over time. Consider two meeting events (i, j, τ) and (i, j', τ') involving the same agent $i \in \mathcal{A}$ and two other agents j and j' (possibly $j = j'$). Then, the distance between the (unknown) locations of the two events is at most $\bar{v}_i|\tau - \tau'|$, since agent i moves at speed at most $\bar{v}_i$, and the time between the two events is $|\tau - \tau'|$.

The *feasibility region* $R(\chi)$ of a meeting event $\chi \in \mathcal{E}_{Meetings}$ is the locus of points (locations) where the meeting χ could have taken place, consistent with all of the known data in the event log. By slightly abusing notation, we will let $R(\tau, i)$ denote the locus of possible locations of agent i at time τ. Of course, if τ coincides with the time of a meeting event $\chi = (i, j, \tau)$ in which agent i is involved, then $R(\tau, i) = R(\chi)$; and if agent i is involved in a GPS event (i, τ, p), then $R(\tau, i) = \{p\}$.

We define the *event graph* $\mathcal{G} = \mathcal{G}(\mathcal{E}_{GPS} \cup \mathcal{E}_{Meetings}, E)$ to be the graph whose vertex set is the set of events $(\mathcal{E}_{GPS} \cup \mathcal{E}_{Meetings})$ and whose edges are defined as follows: Two events χ, χ', with associated times τ and τ', define an edge $(\chi, \chi') \in E$ if they involve a common agent $i \in \mathcal{A}$, and agent i was not involved in any events (meetings or GPS events) at times between τ and τ'. We assign the weight $w(\chi, \chi') = \bar{v}_i|\tau - \tau'|$ to the edge (χ, χ'); the weight of the edge is an upper bound on the distance between the two meeting locations.

4 COMPUTING FEASIBILITY REGIONS

4.1 Preliminaries

Let m be the total number of events. Let the region $\mathbf{R} \subseteq \mathbb{R}^{2m}$ be the locus of all points $(x_1, y_1 \ldots x_m, y_m) \in \mathbb{R}^{2m}$ such that placing event χ_i at (x_i, y_i), for all $i = 1, \ldots, m$, satisfies all position and distance constraints.

Problem 4.1. *Given the event graph $\mathcal{G}$ defined above, compute* $\mathbf{R}$.

Observe that the set $\mathbf{R}$ is defined by a set of convex constraints. Specifically, the GPS events fix the position of a vertex in the graph and thus correspond to fixing two coordinates of all points in $\mathbf{R}$. Each edge (χ, χ') of the graph gives rise to the convex constraint $d((x, y), (x', y')) \leq w(\chi, \chi')$, where (x, y) and (x', y') are the locations of the events χ and χ', respectively. Now we immediately conclude:

Lemma 4.1. $\mathbf{R}$ *is convex.*

Lemma 4.2. $R(\chi)$ *is convex, for every event* χ.

Proof. $R(\chi)$ is the orthogonal projection of $\mathbf{R}$ to the plane. Since $\mathbf{R}$ is convex, so is its projection. □

There are several natural ways to measure distances between points in the plane. Two of them are the Euclidean (L_2) distance $d_2((x, y), (x', y')) = \sqrt{(x - x')^2 + (y - y')^2}$, and the L_∞-distance $d_\infty((x, y), (x', y')) = \max\{|x - x'|, |y - y'|\}$. We suggest several algorithms for solving Problem 4.1. It appears that there are large gaps in the difficulty of solving the problem in the L_2 and L_∞ case. We will start with the latter case, which is easier.

4.2 Computing feasibility regions in L_∞

In this section, the distance between any pair of locations, such as positions of GPS or meeting events or initial locations of agents, is measured using the L_∞ distance. In other words, we are computing the bounding boxes for the feasibility regions. Intuitively, the L_∞ distance allows us to treat the x- and the y-coordinate of the location separately and independently, so it is sufficient to solve two independent one-dimensional problems. We show how to compute the bounding boxes for the feasibility regions of all events $R(\chi)$ for $\chi \in \mathcal{E}_{Meetings}$.

Let $X(R(\chi))$ denote the projection of $R(\chi)$ to the x-axis; define $Y(R(\chi)))$ analogously. As in the proof of Lemma 4.2, both $X(R(\chi))$ and $Y(R(\chi))$ are one-dimensional convex sets, i.e., intervals. Since x- and y-coordinate constraints are independent under the L_∞ distance, $R(\chi) = X(R(\chi)) \times Y(R(\chi))$ is an axis-parallel rectangle. Let $MaxX(\chi)$ be the rightmost point of $X(R(\chi))$. Below we explain how to compute $MaxX(\chi)$. Computing the remaining three coordinates is completely analogous and omitted for brevity. Without loss of generality, we will assume that the x-coordinates of all meetings are non-negative; otherwise the entire input set can be shifted to ensure that this condition holds.

We define a new graph $\mathcal{G}^x(\mathcal{E}_{GPS} \cup \mathcal{E}_{Meetings}, E)$, which is a modification of $\mathcal{G}$.

Algorithm L_∞ *Feasibility*

1. Initialize $\mathcal{G}^x(\mathcal{E}_{GPS} \cup \mathcal{E}_{Meetings}, E)$ to $\mathcal{G}$, with weights as in Section 3.2.
2. Add a new vertex s_0 to $\mathcal{G}^x$, placed at $x = 0$.
3. For every $\chi \in \mathcal{E}_{GPS}$, add to $\mathcal{G}^x$ an edge (s_0, χ), with weight equal to the x-coordinate of the location of the event.
4. Run Dijkstra's algorithm starting at s_0, computing distances $\delta(s_0, \chi)$, which are the lengths of the shortest paths $s_0 \rightsquigarrow \chi$ in $\mathcal{G}^x$.
5. **return** $\delta(s_0, \chi)$ for every χ.

Lemma 4.3. $\delta(s_0, \chi)$ *is* $MaxX(\chi)$*, i.e., the rightmost point of the x-projection of* $R(\chi)$.

Proof. By induction from left to right. We set $MaxX(s_0) = 0$, so the base of the induction is satisfied.

Assume that the claim is true for every event χ' with $MaxX(\chi') < MaxX(\chi)$. Observe that $MaxX(\chi)$ could only be determined by an event χ'' (either a GPS or a meeting event) that is located to the left of $MaxX(\chi)$; otherwise we could move $MaxX(\chi)$ further to its right. By induction hypothesis $MaxX(\chi'') = \delta(s_0, \chi'')$. Hence $MaxX(\chi) = \min_{\chi''}\{MaxX(\chi'') + w(\chi'', \chi)\} = \min_{\chi''}\{\delta(s_0, \chi'') + w(\chi'', \chi)\} = \delta(s_0, \chi)$. □

It is interesting to see another proof of Lemma 4.3. We assume that in every connected component of $\mathcal{G}$ there is at least one GPS event s (with the known coordinate x_s), otherwise only relative positioning information is available and we could add a dummy anchoring GPS event. Determining the largest possible coordinate x_t of an event t is formulated as the linear program (LP)

$$\{\max x_t - x_s : x_i - x_j \leq l_{ij}\},$$

where x_i is the coordinate of event i and l_{ij} is (the right-hand side of) the distance constraint imposed by the speed (as in Section 3.2). The LP's constraint matrix is the incidence matrix of the graph, with 1 and -1 in every row for the tail and the head, respectively,

of each edge, and the dual of this LP is

$$\begin{array}{l}\min \sum l_{ij} y_{ij} \\ \sum_j y_{sj} = 1, \sum_k y_{kt} = 1, \sum_j y_{ij} = \sum_k y_{jk} \forall i \neq s, t,\end{array} \quad (1)$$

which is the min-cost flow LP for sending 1 unit of flow from s to t. By the integrality of the solution (which follows from total unimodularity of the constraint matrix) the flow follows the shortest s-t path. Thus, our problem reduces to shortest path computation.

Theorem 4.1. *Under the L_∞-norm, we can compute $R(\chi)$ for all $\chi \in \mathcal{E}_{Meetings}$, in time $O((m+n)\log n)$, where m is the number of events and n is the number of agents.*

Computing the feasibility region $R(\tau, a)$ of agent a at time τ: Assume that $R(\chi)$ is already computed for every $\chi \in \mathcal{E}_{Meetings}$ using Algorithm L_∞ *Feasibility*. And suppose agent $a \in A$ is not involved in any meeting at time τ. To compute the feasibility region of agent $a \in \mathcal{A}$ at time τ, let χ', χ'' be two meetings involving a immediately before and after τ, at times τ' and τ'', respectively. Then the rightmost point $MaxX(\tau, a)$ of $R(\tau, a)$ occurs at

$$\begin{aligned}\min\{&MaxX(\chi') + (\tau - \tau') \cdot \text{MaxSpeed}(a), \\ &MaxX(\chi'') + (\tau'' - \tau) \cdot \text{MaxSpeed}(a)\}.\end{aligned}$$

As before, the leftmost, topmost, and bottommost points are computed analogously.

Note that, when an attacker computes the feasibility region $R(\tau, a)$ with our algorithm, we assume the attack is generic, i.e., an agent's cloaked/encrypted/suppressed location provides no additional information to this attacker. In practice, an attacker could further reduce the uncertainty by calculating the intersection of its cloaking region and its feasibility region obtained from our algorithm.

4.3 Difficulties with Euclidean (L_2) distances

In this section we present some evidence that the L_2 version of the problem appears to be significantly more challenging.

Hardness in presence of speed lower bounds: We start by proving that when imposing *lower* and *upper* bounds on the speed of the agent motion, even deciding the feasibility of our problem becomes hard.

A *linkage* is an abstract graph with prescribed edge lengths; see, for example [8, 11]. The (two-dimensional) *linkage realizability* (decision) problem is to determine whether a given linkage can be represented as a graph with points in $\mathbb{R}^2$ as vertices and line segments with prescribed Euclidean lengths as edges. The complexity class $\exists\mathbb{R}$ is, roughly speaking, the class of decision problems[1] that can be encoded as a Boolean formula with real variables, usual arithmetic operations, integer constants, and only existential quantifiers ("Does there exist a real solution to the following system of algebraic equations?"). Clearly, checking if $\mathbf{R} = \emptyset$ is such a problem, by definition (see Section 4), so it is in $\exists\mathbb{R}$.

We will argue that the problem of checking if $\mathbf{R} = \emptyset$ is, in fact, $\exists\mathbb{R}$-hard, as it can be reduced from the *unit-length linkage realizability* problem, which is known to be $\exists\mathbb{R}$-complete [26].

Start with a linkage with unit-length edges. Create an agent u for every vertex u (its maximum speed is zero) and an agent (u, v) for every edge (its *minimum and maximum* speed is one). There are no GPS events. We orient every edge arbitrarily. For a directed edge (u, v), we make agent u meet agent (u, v) at time 0 and agent v meet agent (u, v) at time 1. It is easy to check that the original linkage is realizable if and only if for the resulting GPS-and-meeting constraints problem has $\mathbf{R} \neq \emptyset$. So we have reduced the problem of unit-length linkage realizability to the feasibility of our problem with both minimum and maximum speed constraints. Therefore we have the following claim.

Theorem 4.4. *Deciding whether $\mathbf{R} \neq \emptyset$ under L_2 norm is $\exists\mathbb{R}$-complete with both minimum- and maximum-speed constraints.*

Complexity of the boundary of the feasible region. Returning to our feasibility problem with only upper bounds on agents' moving speed, under L_∞ distance the feasibility regions for all agents are axis-parallel rectangles. We note that the L_2 distance constraints are quadratic thus the boundary of $R(\chi)$ can be more complicated. Let $p \in \partial R(\chi)$ be a point on the boundary which corresponds to a realization of the agents' location to satisfy all distance constraints. We say that an edge $(u, w) \in E$ is *taut* if its length equals the maximum distance between its endpoints. Otherwise, it is *loose*.

It is tempting to guess that $R(\chi)$ is a Boolean combination of disks of different radii, and thus bounded by circular arcs. In some examples this is true (see Figure 1 Left), but in general it turns out to be far from the truth.

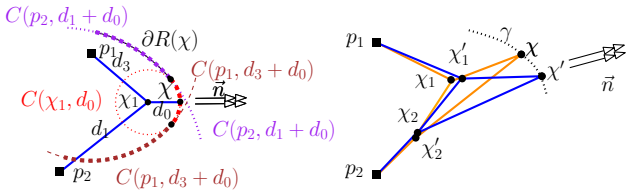


Figure 1: Left: An example where the boundary $\partial R(\chi)$ of $R(\chi)$ under the L_2 distance consists of three circular arcs. Here, p_1, p_2 are locations of GPS events, and there are three distance constraints. At the rightmost boundary segment, all three distance constraints are taut (when χ is pulled in direction $\vec{n}$), while along the other two segments only two of the three are taut. Here, d_0, d_1 and d_2 are the corresponding lengths of the segments $\chi_1\chi, p_2\chi_1$ resp., and $C(p, r)$ is a circle with center p and radius r. Right: Two "snapshots" of a simple configuration that produces a non-circular boundary.

Next we will show a configuration that produces a non-circular boundary. Refer to Fig. 1 Right. The configuration contains two GPS events p_1, p_2 (marked as black boxes) and three meeting events χ, χ_1, χ_2. The length of edges are given. The four events p_1, p_2, χ_1, χ_2 form a quadrilateral with all edges taut. But a quadrilateral is not rigid so it can be moved. The rightmost point of the feasibility region of χ is obtained by pulling in the horizontal direction $\vec{n}$. As we pull χ in directions close to but near $\vec{n}$, the point χ traces out a section of a curve γ that is not a circular arc, as is easy to check.

Notice that this can be replicated as shown in Fig. 2 with two trapezoids sharing an edge. The angles spanned within each trapezoid change as p moves along γ. The placement of the i'th trapezoid depends on the position of the $i-1$'st trapezoid, but since the angles of each of them are changing (as χ slides along γ), the position of no part could be defined as an affine transformation of another part. This suggests that no portion of the configuration moves as

[1] According to Schaefer[26], the best known estimates of where this class lies among other complexity classes is that it contains NP and is contained in PSPACE.

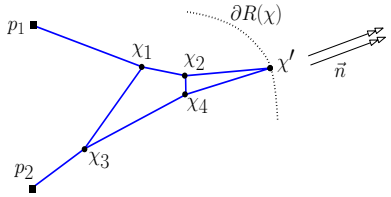


Figure 2: Concatenation of two trapezoids, defined by vertices $p_1p_2\chi_3\chi_1$ and $\chi_1\chi_3\chi_4\chi_2$.

a *rigid* object, hence could not be replaced, even locally, by any smaller subsets of agents. Thus any arc of the boundary could be found only by the solution of a system of $\Theta(n)$ of trapezoids, which suggests high dependencies on the parameters.

4.4 ε-approximation of the L_2 setting

In this section, we approximate the solution to the L_2 problem by measuring the Euclidean distances within a factor of $1+\varepsilon$, where $\varepsilon > 0$ is a user-specified parameter.

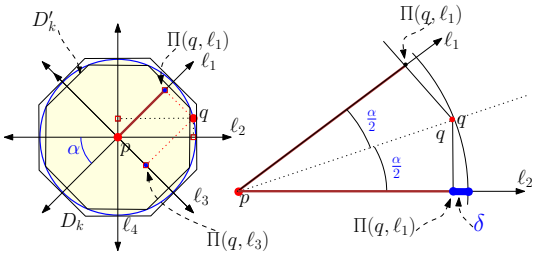


Figure 3: Left: Lines of canonical orientations at spacing α, emerging from point p, and the orthogonal projection $\Pi(q,\ell_i)$ of q on each ℓ_i. If $\|p-q\|_2 = 1$, then the maximum distance $\max_i \|p-\Pi(q,\ell_i)\|$ can be made arbitrarily close to $\|p-q\|$, for small enough α. Right: The discrepancy between the norms is obtained when q is at orientation $\frac{\alpha}{2}$ to p, making the distance δ between them at most $\|p-q\|_2(1-\cos\frac{\alpha}{2})$.

Let $k = k(\varepsilon)$ be an integer to be chosen below. Let $p, q \in \mathbb{R}^2$; refer to Fig. 3. Let $\Phi = \{\ell_1, \ldots, \ell_k\}$ be a set of lines emerging from p and spanning an angle of $\alpha = \pi/k$ between them, such that one of the lines coincides with the x-axis. Consider the regular $2k$-polygon D_k circumscribed around the unit circle centered at p, with each edge perpendicular to a line ℓ_i. For example, if D_2 is the axis-aligned 2×2 square, i.e., the unit ball in the L_∞ metric.

Let $\Pi(q,\ell_i)$ be the orthogonal projection of q to the line ℓ_i. Define the distance $\|p-q\|_{D_k}$ as $\max_i \|p-\Pi(q,\ell_i)\|$. As easily observed this value indicates by which factor should we scale D_k (without shifting its center) so its boundary contains q. Similarly we define the distance $\|p-q\|_{D'_k}$ replacing D_k as the "unit ball" by D'_k, defined as the largest scaled copy of D_k contained inside the unit L_2 disk. See Fig. 3.

It can be verified that for a suitable choice of $k = \Theta(\varepsilon^{-1/2})$

$$\|p-q\|_{D_k} \le \|p-q\|_2 \le \|p-q\|_{D'_k}/(1+\varepsilon).$$

Next let u_i be a unit vector parallel to ℓ_i. We replace the distance constraint $\|p-q\|_2 \le L$ by $\pm(p-q)\cdot u_i \le L$, for all i, thereby introducing a multiplicative error of at most $1+\varepsilon$.

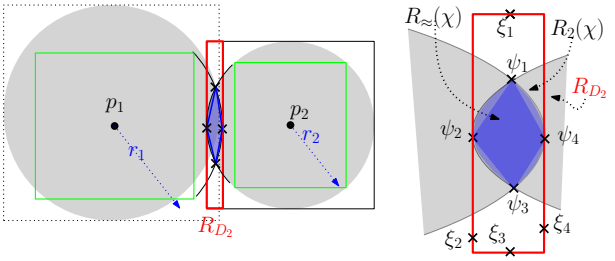


Figure 4: Feasible regions and their approximations for different notion of a distance. Two GPS events are at points p_1 and p_2. Agents leave these locations and meet at χ, traversing a maximum distance of r_1 and r_2, resp. $R_2(\chi)$ is the lens of intersection of the two Euclidean disks. $R_{D_2}(\chi)$ is the red rectangle. $R_{D'_2}(\chi)$ is empty, since the green squares do not intersect. $R_\approx(\chi)$ is the blue diamond.

In this section, we use subscripts on R to indicate the underlying norm. For example, $R_2(\chi)$ denotes the uncertainty region when L_2 norm is used in distance constraints. Section 4.2 showed how to compute $R_\infty(\chi)$ exactly and efficiently.

One approach for approximating $R_2(\chi)$ is to compute well-spread extreme points $\{\psi_1 \ldots \psi_{2k}\}$ on $\partial R_2(\chi)$, along the directions $\pm u_i$, $i = 1,\ldots,k$. Since $R_2(\chi)$ is convex, the convex hull $R_\approx(\chi) = CH(\psi_1 \ldots \psi_k)$ is a good approximation (Fig. 4). Obviously $R_\approx(\chi) \subseteq R_2(\chi)$, and their boundaries are quite close to each other with respect to the Hausdorff Distance: For every point $a \in \partial R_\approx(\chi)$ there is a point $a' \in R_2(\chi)$ within distance $\mathrm{diam}(R_2(\chi))/k$ and vice versa, where $\mathrm{diam}(X)$ is the *diameter* of a compact set X – the maximum distance between any two points in X.

One might wonder what quality of approximation could be obtained by using only Linear Programming (rather than Convex Programming). To follow this approach for approximating $R_2(\chi)$ we solve $2k$ LP problems. In the ith problem we find the extreme point ξ_i of the feasibility region of χ in direction u_i, with distances measured under the L_{D_k} norm. Let $R_{D_k}(\chi) = CH(\{\xi_1 \ldots \xi_{2k}\})$.

Lemma 4.5. *For any event χ,*

$$R_{D'_k}(\chi) \subseteq R_\approx(\chi) \subseteq R_2(\chi) \subseteq R_{D_k}(\chi).$$

To summarize,

- $R_\infty(\chi)$ can be computed in time $O((m+n)\log n)$, for all events χ (Theorem 4.1).
- $R_{D_k}(\chi)$ and $R_{D'_k}(\chi)$ can be computed for a meeting event χ by solving $O(k)$ linear programs with $O((n+m)k)$ constraints each.
- $R_\approx(\chi)$ can be computed by solving $O(k)$ convex programs with $O(n+m)$ constraints each.

It is worth mentioning that from a theoretical point of view, all algorithms mentioned above are polynomial (the first is strongly polynomial, while the others are pseudopolynomial).

Also note that the condition that $\|p-q\|_{D_k} \le L$ for some constant L can be written as $2k$ linear conditions: each of the projections of the vector $p-q$ on the directions u_i does not exceed L. $\|p-q\|_{D'_k} \le L$ is treated similarly, replacing L by $(1-\varepsilon)L$.

Finally, as described above, we compute $R_\approx(\chi)$ as the convex hull of points obtained by solving $2k$ convex programming problems

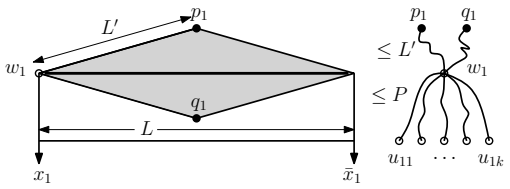


Figure 5: The variable gadget for x_1.

maximizing the projection of the position of χ to each of the $2k$ directions u_i.

Hence we can describe the (approximate version of) $R(\chi)$ as the feasible region of a linear or convex program.

5 DOMAINS WITH OBSTACLES

Next we consider the case where agents may not be able to move along a straight line due to the presence of some obstacles (buildings, lakes or other features of the terrain). This abstract model captures multiple real-life scenarios: (i) Vehicles are confined to paved road. (ii) Pedestrians walk along sidewalks, and in general will not enter building unless these are their final destinations. So even if we obtain bounds of the maximum speed of each agent, the feasibility of locations of meeting points depend on distances along (shortest) paths that avoids the obstacles, rather than crow-fly distances.

In this section we first show that the presence of obstacles makes the problem of testing whether the feasibility region is empty hard, for any L_p metric. Then we show approximate solutions when the obstacles are 'fat' under L_∞ metric.

5.1 Hardness

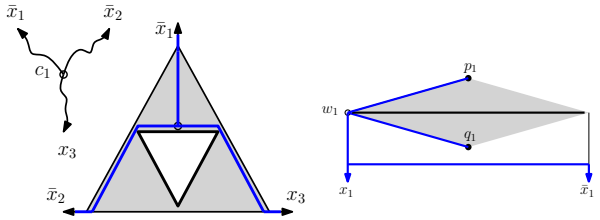
Theorem 5.1. *Testing feasibility of the localization problem in a domain with holes, i.e., checking if* $\mathbf{R} = \emptyset$*, is NP-hard.*

Proof. The proof is by reduction from 1-in-3SAT, in which there are n Boolean variables and m clauses, each with exactly three literals, and we ask whether there is an assignment such that exactly one literal in each clause is true.

For each variable we create a variable gadget. Figure 5 shows an example for variable x_1; we use a solid dot to show a GPS event and a circle to show a meeting event. A variable gadget for x_1 is composed of two GPS events, p_1, q_1, meaning agent p located at p_1 and agent q at location q_1 at time t. These two GPS events are the top and bottom vertices, respectively, of a diamond-shaped room (in gray). This room has two other vertices at distance L' from p_1, q_1; these are two exits of the room. There is a meeting event w_1 at time $t + L'$ for agents p, q. Thus, the meeting event is within distance L' of p_1, q_1. Inside the diamond there is a horizontal segment obstacle, extending almost to the exits, such that the only possible position for w_1 is at either exit of the diamond polygon.

The left and right vertices of the diamond are connected by extremely narrow corridors (of width essentially zero). The two vertical corridors from the left and right exits are connected by another equally narrow corridor of length L, called a *bridge*. See the picture for the illustration. There are two "tendrils" out of each variable gadget, the left one corresponding to x_1 and the right one corresponding to $\overline{x_1}$. The tendril for x_1 (or $\overline{x_1}$) will connect to the clause gadgets that contain x_1 (or $\overline{x_1}$).

For each clause we build a clause gadget, which consists of a special triangular room and one single meeting event. See Figure 6a. We have a regular triangular room of side length R with a regular triangular obstacle inside. The triangular obstacle has side length nearly $R/2$. That is, there are three narrow passages near the three vertices of the triangular obstacles. The three vertices of the triangular room correspond to the three literals of this clause and are connected by narrow corridors to the corresponding tendril of the variable appearing in this clause. If the variable x_1 or its negation $\bar{x}_1$ appears in the ith clause, the meeting event c_i involves agent 1 and happens at time $t + L'_1 + P$. Thus c_i is at most distance P away from the meeting event w_1.



(a) The clause gadget for $\bar{x}_1 \wedge \bar{x}_2 \wedge \bar{x}_3$

(b) The feasible embedding for $x_1 = 0$.

Figure 6: The clause gadget and variable gadget.

Figure 6b shows an example of one possible embedding of the variable gadget, corresponding to the case of $x_1 = 0$. In this case, the meeting event w_1 is located at the left vertex of the gadget such that the distance from w_1 to the meeting event from a clause c_i containing $\overline{x_1}$ has to travel through the horizontal bridge of extra length L. This limits the placement of the meeting event c_i due to the upper bound on distance.

We choose the lengths P, L, and R in a manner that allows precisely three possible positions for each meeting event c_j, at the three midpoints of the edges of the triangular obstacle. Figure 6a gives one placement. This placement makes the meeting point be closer to the exit corresponding to $\overline{x_1}$ and further away from the exits corresponding to $\overline{x_2}$ and x_3. In this case $\overline{x_1} = 1$, meaning the agent 1 travels from meeting location w_1 to the meeting location c_j through the bridge of length L. To make this happen, we just set $L = \frac{3-\sqrt{3}}{4}R$, when distance is measured by the L_2 metric (for other metric spaces the value of R can be properly adjusted).

Last we note that we can place the entire arrangement in the plane. We can add little nooks on the corridors to make their length as prescribed. The narrow corridors connecting the variable gadgets and clause gadgets can cross but these will not affect the statements above due to our design of the length constraints. □

5.2 Fat obstacles

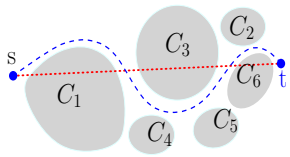
Despite the computational hardness pointed out by the previous section, an adversary might still be able to approximate an agent's location in many environments, as long as their obstacles exhibit geometric property referred to as κ-*fatness*. Roughly speaking, a convex object C in the plane is κ-*fat* (for some fixed constant $\kappa > 0$) if, for each disk D centered on the boundary of C and not fully containing C, the area of $C \cap D$ is at least $\kappa \cdot \text{area}(D)$ [9]. That is, very long and skinny objects will have a very large enclosing disk, but only tiny portions of the disk's area is occupied by C.

Many common, man-made obstacles exhibit such fatness. Of course, certain natural obstructions (rivers, canyons) and man-made structures (fences, train tracks) do not constitute fat obstacles; we leave further study of extensions to (possibly a small number of) non-fat obstacles to future work. In the rest of this section, we assume that all obstacles are κ-fat.

We next show that the algorithm of Section 4.2 could be used to obtain a constant factor approximation on the feasibility regions, as formulated below. First we need the following lemma, proven by Chew *et al* [6].

Lemma 5.1 ([6]). *Let C be a fat obstacle. Let p, q be two points on its boundary, ∂C, and let $\delta = \min\{|\partial_{CCW}(C)|, |\partial_{CW}(C)|\}$, where $\partial_{CCW}(C)$ (resp., $\partial_{CW}(C)$) is the counterclockwise (resp., clockwise) portion of ∂C between p and q, and $|\partial_{CCW}(C)|, |\partial_{CW}(C)|$ are the lengths of these portions. Then, $\|p - q\| \leq K\delta$, where K is a constant that depends only on the fatness parameters.*

The term K is called the *stretch factor*. It can be argued that the shortest path between any two points s, t in the domain with fat obstacles is no longer than $K \cdot |st|$ where $|st|$ is the Euclidean distance between s and t. See the Figure to the right, showing the relatively small discrepancy between the shortest path $s \rightarrow t$ that avoid all obstacles, vs. the length of the segment $|st|$. Specifically, we can take the intersections of the straight line segment st with the fat obstacles, denoted by p_i, q_i with the ith obstacle. Now we build an alternative path that is composed of pieces of segments on st and the shorter paths along the obstacle i to move from p_i to q_i. This path has length at most $K \cdot |st|$. This implies that if we run the Algorithm L_∞ *Feasibility* with the original weights in an obstacle-free setting, we would obtain a region $R'(\chi)$ containing $R(\chi)$, where $R(\chi)$ is the actual feasibility region of χ computed when taking all obstacles into account. In addition, we could compute the stretch factor K of all obstacles, replace the length $w(u, v)$ of each edge of $\mathcal{G}$ by $w(u, v)/K$, and recalculate the uncertainty region for each χ, resulting in new regions, denoted $R''(\chi)$, contained within $R(\chi)$. In summary,



Theorem 5.2. *In time $O((m+n)\log n)$ we can compute feasibility regions $R'(\chi), R''(\chi)$ for the events such that $R''(\chi) \subseteq R(\chi) \subseteq R'(\chi)$, when the domain has fat obstacles and the distance is measured by shortest path in the domain, under the L_∞ metric.*

6 EVALUATION

In this section we evaluate how the algorithms work on both a synthetic dataset and a real trajectory dataset. All experiments are run on a laptop with Intel(R) Core(TM) i5-4200M CPU @ 2.50GHZ and 4 GB memory.

6.1 Simulation on Synthetic Datasets

We first present the simulation results of our L_∞ feasibility algorithm in Section 4.2 on synthetic datasets. We assume that some agents share their locations, while others are concerned about their location privacy. We refer to the latter ones as *privacy-aware agents*, or *PA agents*, for short. The others are *non-PA* agents. We simulate the movements of all agents using Processing [1], assuming a simple Markovian model. Each agent moves randomly along the x-axis in an interval of length 800 meters, with a random speed between 2 meters per time frame and 4 meters per time frame. Every non-PA agent reports a GPS event once her location is updated, and all agents (PA and non-PA) report their meeting events with others with a given frequency (e.g., once every 5 time frames).

In our experiment, we have 12 agents. First, we test the impact of the number of non-PA agents on the average uncertainty of PA agents. The average uncertainty of a PA agent is calculated as

$$\frac{\sum_{\#\tau} \sum_{\#a} R(\tau, a)}{\#\tau \cdot \#a},$$

where a is a PA agent and $R(\tau, a)$ is the width of its feasibility region at time frame τ. Having a small location uncertainty is undesirable, as it compromises an agent's location.

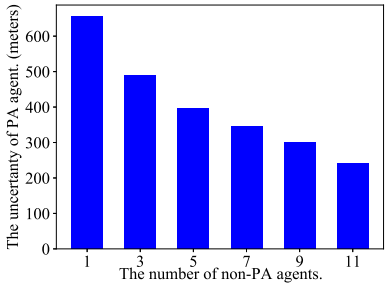


Figure 7: The impact of the number of non-PA agents on the average uncertainty of PA agents.

The simulation result is demonstrated in Fig. 7. As expected, as more agents report GPS events, the average size of the feasibility regions decreases. Table 1 shows the average number of GPS events collected in our experiment.

Table 1: # GPS events

#Non-PA agents	1	3	5	7	9	11
#GPS events	13	41	64	103	128	166

In Fig. 8 we show the average uncertainty of a PA agent as a function of the meeting report frequency, defined to be the number of time frames between two consecutive meetings. It is interesting to see the correlation between meeting report frequency and the average uncertainty region size. In the same period of time, with the shorter meeting report frequency (i.e., fewer time frames between meetings), more meeting events are reported and collected (see Table 2). As a result, the more reported meeting events during the same time period, the smaller the average uncertainty of PA agents.

Table 2: # Meeting events

#time frames	5	10	15	20	25
#meeting events	366	183	129	95	59

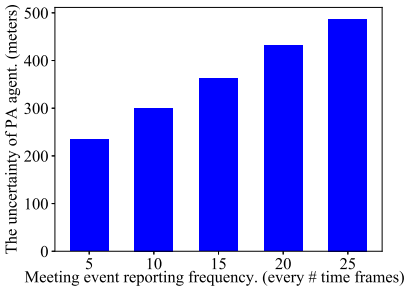


Figure 8: The impact of meeting report frequency on the average uncertainty of PA agent.

6.2 Experiment on real GPS trajectories

We further evaluate our algorithm with the trajectories of 6,099 taxis in Shenzhen [12], sampled every five minutes during one hour. Each trajectory is sampled by 13 GPS locations. All of the GPS locations are within the range of longitude from $113.8°E$ to $114.3°E$, and latitude from $22.45°N$ to $22.75°N$, which is an area of about $1,847km^2$. To simulate the meeting events, we assume that each agent travels along the line segment between two consecutive GPS locations with uniform speed. If two agents arrive to the same position within a time interval of one second, we record it as a meeting event. There are 14,534 meeting events in this dataset.

Again, we run the L_∞ feasibility algorithm and investigate the impact of the number of GPS events on the results. First, we let all vehicles report GPS events with different frequencies. We calculate the feasibility regions of all meeting events. In other words, there are no PA agents in this case, but the number of GPS events changes with the GPS reporting frequency. As shown in Fig. 9, when each vehicle reports one GPS event every five minutes, the average width of the feasibility regions is only 711 meters while its average height is 973 meters. Even if each vehicle only reports two GPS events at the start and end of the duration, i.e., with frequency of reporting one GPS every 60 minutes, the average size of the feasibility regions can be narrowed down to 10 kilometers in width and 12 kilometers in height, which is still relatively small compared to the area of the entire range of motion of the vehicles.

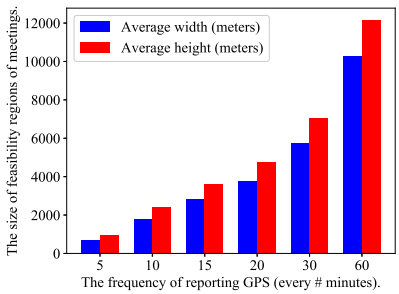


Figure 9: The impact of GPS reporting frequency on the average size of feasibility regions of meeting events.

Then we increase the number of PA agents, which are randomly chosen from the 6,099 vehicles. The others still report their GPS events once every 5 minutes. As illustrated in Fig. 10, our algorithm works effectively with the existence of PA agents. For instance, even with about one third of the vehicles (i.e. 2,000 vehicles) not reporting their GPS locations, both the average width and the average height of the feasibility regions are less than 1,600 meters. Also, the average size of the feasibility regions increases as fewer vehicles report GPS events, which is consistent with our previous simulation results on synthetic datasets (i.e., the results shown in Fig. 7).

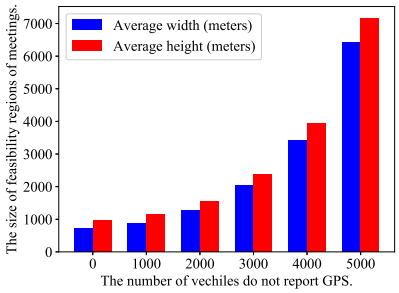


Figure 10: The impact of PA agents on the average size of feasibility regions of meeting events.

Both Fig. 9 and Fig. 10 demonstrate that the more GPS events collected, the more accurate our algorithm performs. More importantly, even with a small number of GPS events, our algorithm can still derive a small feasibility region for each meeting event.

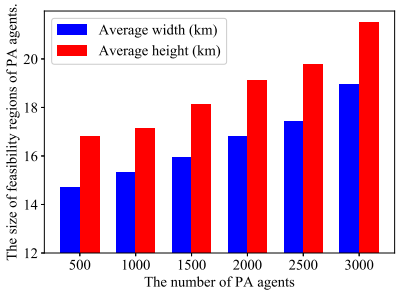


Figure 11: The impact of PA agents on the average size of feasibility regions of agents.

The above results have shown the capability of our L_∞ feasibility algorithm to infer the locations of meeting events. As discussed in Section 4.2, we can further calculate a PA agent a's feasibility region at time τ from the feasibility regions of two meetings involving a immediately before and after τ, even if a is not involved in a meeting at τ. We also implement this method to recover all PA agents' feasibility regions at the midpoint of the hour, and present the impact of the number of PA agents on the results. Since there are only 3,477 vehicles involved in at least two meetings before and after the midpoint time, respectively, we randomly choose the PA agents from these 3,477 vehicles. Fig. 11 shows the average size of all PA agents' feasibility regions at the midpoint time, which also grows with the increase in the number of PA agents. Compared

to the results in Fig. 10, the feasibility regions of PA agents are larger. This is due to the sparsity of meeting events in our dataset (recall that we have shown the impact of the number of meetings in Fig. 8). However, they are still much less than the whole area of the dataset. Thus, we can conclude that our algorithm is also effective in revealing agents' feasibility regions.

7 CONCLUSION AND FUTURE WORK

In this paper, we focused on understanding the information leakage of a user's location trajectory caused by other privacy-insensitive users' social behaviors, such as posting a meeting with location data. To the best of our knowledge, this is the first work to quantify the limits of location privacy for sporadic location traces, without assuming specific mobility models. Our model is general enough to capture a wide range of location privacy protection techniques. So far we have assumed that when two agents meet their meeting time is recorded precisely. The algorithm using L_∞ can be easily extended if a time interval is associated with each meeting event.

The obvious open question is how to defend against this type of attack. Common solutions to this problem are based on changing some of the reported locations (e.g., via random perturbation or cloaking) or filtering out some of them; however, this action risks upsetting users who want to publish their locations, at an accuracy they wish to determine. In addition, such methods will degrade the usability of the LBS/LBSN. To address this issue, we will explore the approach of adding fake (dummy) agents and trajectories, which are indistinguishable from real users' trajectories, in order to make it computationally intractable to recover real agents' uncertainty regions.

ACKNOWLEDGMENTS

Work on this paper by B. Aronov has been partially supported by the National Science Foundation (CCF-11-17336, CCF-12-18791, and CCF-15-40656), and by the US-Israel Binational Science Foundation (project 2014/170). Work by J. Gao and J. Ding has been partially supported by the National Science Foundation (CCF-1535900, CNS-1618391, and DMS-1737812). Work by M. Li has been partially supported by the National Science Foundation (CNS-1731164). Work by J. Mitchell has been partially supported by the National Science Foundation (CCF-1526406) and by the US-Israel Binational Science Foundation (project 2016/116). Work by V. Polishchuk has been partially supported by the Swedish Transport Administration.

REFERENCES

[1] 2001. Processing. https://processing.org/.
[2] Miguel E Andrés, Nicolás E Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. 2013. Geo-indistinguishability: Differential privacy for location-based systems. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, 901–914.
[3] George Argyros, Theofilos Petsios, Suphannee Sivakorn, Angelos Keromytis, and Jason Polakis. 2017. Evaluating the Privacy Guarantees of Location Proximity Services. *ACM Transactions on Privacy and Security* (2017).
[4] Vincent Bindschaedler and Reza Shokri. 2016. Synthesizing plausible privacy-preserving location traces. In *Security and Privacy (SP), 2016 IEEE Symposium on*. IEEE, 546–563.
[5] Vincent Bindschaedler, Reza Shokri, and Carl A Gunter. 2017. Plausible deniability for privacy-preserving data synthesis. *Proceedings of the VLDB Endowment* 10, 5 (2017), 481–492.
[6] L. Paul Chew, Haggai David, Matthew J. Katz, and Klara Kedem. 2002. Walking Around Fat Obstacles. *Inf. Process. Lett.* 83, 3 (Aug. 2002), 135–140. https://doi.org/10.1016/S0020-0190(01)00321-0
[7] Benny Chor, Oded Goldreich, Eyal Kushilevitz, and Madhu Sudan. 1995. Private information retrieval. In *Proceedings of 36th Annual Symposium on the Foundations of Computer Science*. IEEE, 41–50.
[8] Robert Connelly and Erik D. Demaine. 2004. Geometry and Topology of Polygonal Linkages. In *Handbook of Discrete and Computational Geometry, Second Edition*. 197–218. https://doi.org/10.1201/9781420035315.ch9
[9] Mark de Berg, A Frank van der Stappen, Jules Vleugels, and Matthew J Katz. 2002. Realistic input models for geometric algorithms. *Algorithmica* 34, 1 (2002), 81–97.
[10] Yves-Alexandre De Montjoye, César A Hidalgo, Michel Verleysen, and Vincent D Blondel. 2013. Unique in the crowd: The privacy bounds of human mobility. *Scientific reports* 3 (2013), 1376.
[11] Erik D Demaine and Joseph O'Rourke. 2007. *Geometric Folding Algorithms*. Cambridge University Press, Cambridge.
[12] Jiaxin Ding, Jie Gao, and Hui Xiong. 2015. Understanding and modelling information dissemination patterns in vehicle-to-vehicle networks. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 41.
[13] Cynthia Dwork and Aaron Roth. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science* 9, 3–4 (2014), 211–407.
[14] Gabriel Ghinita, Maria Luisa Damiani, Claudio Silvestri, and Elisa Bertino. 2016. Protecting Against Velocity-Based, Proximity-Based, and External Event Attacks in Location-Centric Social Networks. *ACM Trans. Spatial Algorithms Syst.* 2, 2, Article 8 (June 2016), 36 pages. https://doi.org/10.1145/2910580
[15] Gabriel Ghinita, Panos Kalnis, Ali Khoshgozaran, Cyrus Shahabi, and Kian-Lee Tan. 2008. Private queries in location based services: anonymizers are not necessary. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. ACM, 121–132.
[16] Marco Gruteser and Drik Grunwald. 2003. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In *Proc. of 1st International Conference on Mobile Systems, Applications and Services (MobiSys'03)*.
[17] Panayiotis Kotzanikolaou, Constantinos Patsakis, Emmanouil Magkos, and Michalis Korakakis. 2016. Lightweight private proximity testing for geospatial social networks. *Computer Communications* 73 (2016), 263–270.
[18] John Krumm. 2007. Inference attacks on location tracks. *Pervasive computing* (2007), 127–143.
[19] Muyuan Li, Haojin Zhu, Zhaoyu Gao, Si Chen, Le Yu, Shangqian Hu, and Kui Ren. 2014. All your location are belong to us: Breaking mobile social networks for automated user location tracking. In *Proceedings of the 15th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 43–52.
[20] Takao Murakami. 2017. Expectation-Maximization Tensor Factorization for Practical Location Privacy Attacks. In *Proc. of Privacy Enhancing Technologies (PETS)*.
[21] Arvind Narayanan, Narendran Thiagarajan, Mugdha Lakehani, Mike Hamburg, and Dan Boneh. 2011. Location Privacy via Private Proximity Testing. In *Proceedings of the Network and Distributed Security Symposium (NDSS)*.
[22] Ben Niu, Qinghua Li, Xiaoyan Zhu, Guohong Cao, and Hui Li. 2014. Achieving k-anonymity in Privacy-Aware Location-Based Services. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*.
[23] Pierangela Samarati and Latanya Sweeney. 1998. *Protecting Privacy when Disclosing Information: k-Anonymity and Its Enforcement through Generalization and Suppression*. Technical Report. SRI International.
[24] Iasonas Polakis, George Argyros, Theofilos Petsios, Suphannee Sivakorn, and Angelos Keromytis. 2015. Where is Wally? Precise User Discovery Attacks in Location Proximity Services. In *Proc. of CCS'15*.
[25] Apostolos Pyrgelis, Carmela Troncoso, and Emiliano De Cristofaro. 2017. What Does The Crowd Say About You? Evaluating Aggregation-based Location Privacy. In *Proc. of Privacy Enhancing Technologies (PETS)*. 76–96.
[26] Marcus Schaefer. 2013. Realizability of Graphs and Linkages. In *Thirty Essays on Geometric Graph Theory*, János Pach (Ed.). Springer New York, New York, NY, 461–482. https://doi.org/10.1007/978-1-4614-0110-0_24
[27] Reza Shokri, George Theodorakopoulos, George Danezis, Jean-Pierre Hubaux, and Jean-Yves Le Boudec. 2011. Quantifying Location Privacy: The Case of Sporadic Location Exposure. In *Proc. of Privacy Enhancing Technologies (PETS)*.
[28] Reza Shokri, George Theodorakopoulos, Jean-Pierre Hubaux, and Jean-Yves Le Boudec. 2011. Quantifying Location Privacy. In *Proc. of IEEE S&P'11*.
[29] M. Veyunpublishedtsman. 2014. How I was able to track the location of any Tinder user. http://blog.includesecurity.com/2014/02/how-i-was-able-to-track-location-of-any.html.
[30] Boyang Wang, Ming Li, and Haitao Wang. 2016. Geometric range search on encrypted spatial data. *IEEE Transactions on Information Forensics and Security* 11, 4 (2016), 704–719.
[31] Boyang Wang, Ming Li, Haitao Wang, and Hui Li. 2015. Circular range search on encrypted spatial data. In *Communications and Network Security (CNS), 2015 IEEE Conference on*. IEEE, 182–190.
[32] David J Wu, Joe Zimmerman, Jérémy Planul, and John C Mitchell. 2016. Privacy-preserving shortest path computation. (2016). arXiv preprint arXiv:1601.02281.
[33] Hao Wu and Yih-Chun Hu. 2016. Location Privacy with Randomness Consistency. In *Proceedings on Privacy Enhancing Technologies*.
[34] Yonghui Xiao and Li Xiong. 2015. Protecting Locations with Differential Privacy Under Temporal Correlations. In *CCS*. 1298–1309.
[35] Man Lung Yiu, Christian S Jensen, Xuegang Huang, and Hua Lu. 2008. Spacetwist: Managing the trade-offs among location privacy, query performance, and query

accuracy in mobile services. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*. IEEE, 366–375.

[36] Yao Zheng, Ming Li, Wenjing Lou, and Thomas Hou. 2015. Location Based Handshake and Private Proximity Test with Location Tags. *IEEE Transactions on on Dependable and Secure Computing* (2015).

[37] Ge Zhong, Ian Goldberg, and Urs Hengartner. 2007. Louis, Lester and Pierre: Three Protocols for Location Privacy. In *Proc. of Privacy Enhancing Technologies (PETS)*.