Optimal Geo-Indistinguishable Mechanisms for Location Privacy

Nicolás E. Bordenabe
INRIA and École Polytechnique
nbordenabe@lix.polytechnique.fr

Konstantinos Chatzikokolakis
CNRS and École Polytechnique
kostas@lix.polytechnique.fr

Catuscia Palamidessi
INRIA and École Polytechnique
catuscia@lix.polytechnique.fr

ABSTRACT

We consider the geo-indistinguishability approach to location privacy, and the trade-off with respect to utility. We show that, given a desired degree of geo-indistinguishability, it is possible to construct a mechanism that minimizes the service quality loss, using linear programming techniques. In addition we show that, under certain conditions, such mechanism also provides optimal privacy in the sense of Shokri et al. Furthermore, we propose a method to reduce the number of constraints of the linear program from cubic to quadratic, maintaining the privacy guarantees and without affecting significantly the utility of the generated mechanism. This reduces considerably the time required to solve the linear program, thus enlarging significantly the location sets for which the optimal mechanisms can be computed.

Categories and Subject Descriptors

C.2.0 [**Computer–Communication Networks**]: General—*Security and protection*; K.4.1 [**Computers and Society**]: Public Policy Issues—*Privacy*

Keywords

Location privacy; Location obfuscation; Geo-indistinguishability; Differential privacy; Linear optimization

1. INTRODUCTION

While location-based systems (LBSs) have demonstrated to provide enormous benefits to individuals and society, these benefits come at the cost of users' privacy: as discussed in [1, 2, 3], location data can be easily linked to a variety of other information about an individual, and expose sensitive aspects of her private life such as her home address, her political views, her religious practices, etc.. There is, therefore, a growing interest in the development of location-privacy protection mechanisms (LPPMs), that allow to use LBSs while providing sufficient privacy guarantees for the user. Most of the approaches in the literature are based on perturbing the information reported to the LBS provider, so to prevent the disclosure of the user's location [4, 5, 6, 7, 8, 9].

Clearly, the perturbation of the information sent to the LBS provider leads to a degradation of the quality of service, and consequently there is a trade-off between the level of privacy that the user wishes to guarantee and the service quality loss (QL) that she will have to accept. The study

of this trade-off, and the design of mechanisms which optimize it, is an important research direction started with the seminal paper of Shroki et al. [10].

Obviously, any such study must be based on meaningful notions of privacy and of quality loss. The authors of [10] consider the privacy threats deriving from a Bayesian adversary. More specifically, they assume that the adversary knows the prior probability distribution on the user's possible locations, and they quantify privacy as the expected error, namely the expected distance between the true location and the best guess of the adversary once she knows the location reported to the LBS. We refer to this quantity as ADVERROR. The adversary's guess takes into account the information already in her possession (the prior probability), and it is by definition more accurate, in average, than the reported location. We also say that the adversary may *remap* the reported location.

The notion of quality loss adopted in [8] is also defined in terms of the expected distance between the real location and the reported location, with the important difference that the LBS is not assumed to know the user's prior distribution (the LBS is not tuned for any specific user), and consequently it does not apply any remapping. Note that the notion of distance used for expressing QL does not need to be the same as the one used to measure location privacy. When these two notions coincide, then QL is always greater than or equal to the location privacy, due to the fact that the adversary can make use of the prior information to her advantage. The optimal mechanism of [8] is defined as the one which maximizes privacy for a given QL threshold, and since these measures are linear functions of the noise (characterized by the conditional probabilities of each reported location given a true location), such mechanism can be computed by solving a linear optimization problem.

In this paper, we consider the geo-indistinguishability framework of [9], a notion of location privacy based on differential privacy [11], and more precisely, on its extension to arbitrary metrics proposed in [12]. Intuitively, a mechanism provides geo-indistinguishability if two locations that are geographically close have similar probabilities to generate a certain reported location. Equivalently, the reported location will not increase by much the adversary's chance to distinguish the true location among the nearby ones. Note that this notion protects the accuracy of the location: the adversary is allowed to distinguish locations which are far away. It is important to note that the property of geo-indistinguishability does not depend on the prior. This is a

feature inherited from differential privacy, which makes the mechanism robust with respect to composition of attacks in the same sense as differential privacy.

We study the problem of optimizing the trade-off between geo-indistinguishability and quality of service. More precisely, given a certain threshold on the degree of geo-indistinguishability, and a prior, we aim at obtaining the mechanism K which minimizes QL. Thanks to the fact that the property of respecting the geo-indistinguishability threshold can be expressed by linear constraints, we can reduce the problem of producing such a K to a linear optimization problem, which can then be solved by using standard techniques of linear programming.

It should be remarked that our approach is, in a sense, dual wrt the one of [8]. The latter fixes a bound on QL and optimizes the location privacy. Here, on the contrary, we fix a bound on the location privacy and then optimize QL. Another important difference is that in [8] the privacy degree of the optimal mechanism, measured by ADVERROR, is guaranteed for a specific prior only, while in our approach the privacy guarantee of the optimal mechanism is in terms of geo-indistinguihability, which does not depend on the prior. In our opinion, this is an important feature of the present approach, as it is difficult to control the prior knowledge of the adversary. Consider, for instance, a user for which the optimal mechanism has been computed with respect to his average day (and consequent prior π), and who has very different habits in the morning and in the afternoon. By simply taking into account the time of the day, the adversary gains some additional knowledge that determines a different prior, and the privacy guarantees of the optimal mechanism of [8] can be severely violated when the adversary uses a prior different from π.

However, when the notion of distance used to measure the QL coincides with that used for expressing the degree of privacy according to ADVERROR, then, somewhat surprisingly, our optimal mechanism K turns out to be also optimal in terms of ADVERROR, in a sense getting the best of both approaches. Intuitively, this is due to the fact that the property of geo-indistinguishability is not affected by remapping. Hence, the expected error of the adversary must coincide with QL, i.e., the adversary cannot gain anything by any remapping H, or otherwise KH would be still geo-indistinguishable and provide a better QL. Since privacy coincides with the QL, it must also be optimal. In conclusion, we obtain a geo-indistinguishable K with minimum QL and maximum degree of privacy (for that QL).

Note that the optimal mechanisms are not unique, and ours does not usually coincide with the one produced by the algorithm of [8]. In particular the one of [8] in general does not provide geo-indistinguishability, while ours does, by design. The robustness of the geo-indistinguishability property seems to affect favorably also other notions of privacy: We have evaluated the two mechanisms with the privacy definition of [8] on two real datasets, and we have observed that, while the mechanism of [8] by definition offers the best privacy on the prior for which it is computed, ours can perform significantly better when we consider different priors.

We now turn our attention to efficiency concerns. Since the optimal mechanism is obtained by solving a linear optimization problem, the efficiency depends crucially on the number of constraints used to express geo-indistinguishability. We note that this number is, in general, cubic with respect

to the amount of locations considered. We show that we are able to reduce this number from cubic to quadratic, using an approximation technique based on constructing a suitable spanning graph of the set of locations. The idea is that, instead of considering the geo-indistinguishability constraints for every pair of locations, we only consider those for every edge in the spanning graph. We also show, based on experimental results, that for a reasonably good approximation our approach offers an improvement in running time with respect to method of Shokri et al. We must note however that the mechanism obtained this way is no longer optimal with respect to the original metric, but only with respect to the metric induced by the graph, and therefore the QL of the mechanism might be higher, although our experiments also show that this increase is not significant.

Note that in this paper we focus on the case of *sporadic* location disclosure, that is, we assume that there is enough time between consecutive locations reported by the user, and therefore they can be considered independent. Geo-indistinguishability can be applied also in case of correlation between consecutive points, but additional care must be taken to avoid the degradation of privacy, that could be significant when the number of consecutive locations is high. The problem of correlation is orthogonal to to the goals of this paper. We refer to [13] for a study of this problem.

Contribution.

The main contributions of this paper are the following:

- We present a method based on linear optimization to generate a mechanism that is geo-indistinguishable and achieves optimal utility. Furthermore when the notions of distance used for QL coincide with that used for geo-indistinguishability, then the mechanism is also optimal with respect to the expected error of the adversary.

- We evaluate our approach under different priors (generated from real traces of two widely used datasets), and show that it outperforms the other mechanisms considered.

- We propose an approximation technique, based on spanning graphs, that can be used to reduce the number of constraints of the optimization problem and still obtain a geo-indistinguishable mechanism.

- We measure the impact of the approximation on the utility and the number of constraints, and analyze the running time of the whole method, obtaining favorable results.

Plan of the paper.

The rest of the paper is organized as follows. Next section recalls some preliminary notions. In Section 3 we illustrate our method to produce a geo-indistinguishable and optimal mechanism as the solution of a linear optimization problem, and we propose a technique to reduce the number of constraints used in the problem. In Section 4 we evaluate our mechanism with respect to other ones in the literature. Finally, in Section 5, we discuss related work and conclude.

This paper is the report version of a work that appeared in the proceedings of the 21st ACM Conference on Computer Security. Scottsdale, Arizona, USA, Nov. 2014 (CCS'14).

2. PRELIMINARIES

2.1 Location obfuscation, quality loss and adversary's error

A common way of achieving location privacy is to apply a *location obfuscation* mechanism, that is a probabilistic function $K : \mathcal{X} \to \mathcal{P}(\mathcal{X})$ where $\mathcal{X}$ is the set of possible locations, and $\mathcal{P}(\mathcal{X})$ denotes the set of probability distributions over $\mathcal{X}$. K takes a location x as input, and produces a *reported location* z which is communicated to the service provider. In this paper we generally consider $\mathcal{X}$ to be finite, in which case K can be represented by a stochastic matrix, where k_{xz} is the probability to report z from location x.

A prior distribution $\pi \in \mathcal{P}(\mathcal{X})$ on the set of locations can be viewed either as modelling the behaviour of the user (the *user profile*), or as capturing the adversary's *side information* about the user. Given a prior π and a metric d on $\mathcal{X}$, the expected distance between the real and the reported location is:

$$\textsc{ExpDist}(K, \pi, d) = \sum_{x,z} \pi_x k_{xz} d(x, z)$$

From the user's point of view, we want to quantify the service *quality loss (QL)* produced by the mechanism K. Given a *quality metric* d_Q on locations, such that $d_Q(x, z)$ measures how much the quality decreases by reporting z when the real location is x (the Euclidean metric d_2 being a typical choice), we can naturally define the quality loss as the expected distance between the real and the reported location, that is $\text{QL}(K, \pi, d_Q) = \textsc{ExpDist}(K, \pi, d_Q)$. The QL can also be viewed as the (inverse of the) utility of the mechanism.

Similarly, we want to quantify the *privacy* provided by K. A natural approach, introduced in [10] is to consider a Bayesian adversary with some prior information π, trying to remap z back to a guessed location $\hat{x}$. A remapping strategy can be modelled by a stochastic matrix H, where $h_{z\hat{x}}$ is the probability to map z to $\hat{x}$. Then the privacy of the mechanism can be defined as the expected error of an adversary under the best possible remapping:

$$\textsc{AdvError}(K, \pi, d_A) = \min_H \textsc{ExpDist}(KH, \pi, d_A)$$

Note that the composition KH of K and H is itself a mechanism. Similarly to d_Q, the metric $d_A(x, \hat{x})$ captures the adversary's loss when he guesses $\hat{x}$ while the real location is x. Note that d_Q and d_A can be different, but the canonical choice is to use the Euclidean distance for both.

A natural question, then, is to construct a mechanism that achieves *optimal privacy*, given a *QL constraint*.

DEFINITION 1. *Given a prior π, a quality metric d_Q, a quality bound q and an adversary metric d_A, a mechanism K is q-$\textsc{OptPriv}(\pi, d_A, d_Q)$ iff*

1. *$\text{QL}(K, \pi, d_Q) \leq q$, and*

2. *for all mechanisms K', $\text{QL}(K', \pi, d_Q) \leq q$ implies $\textsc{AdvError}(K', \pi, d_A) \leq \textsc{AdvError}(K, \pi, d_A)$*

In other words, a q-$\textsc{OptPriv}$ mechanism provides the best privacy (expressed in terms of $\textsc{AdvError}$) among all mechanisms with QL at most q. This problem was studied in [8], providing a method to construct such a mechanism for any q, π, d_A, d_Q, by solving a properly constructed linear program.

2.2 Differential privacy

Differential privacy was originally introduced in the context of statistical databases, requiring that a query should produce similar results when applied to *adjacent* databases, i.e. those differing by a single row. The notion of adjacency is related to the Hamming metric $d_h(x, x')$ defined as the number of rows in which x, x' differ. Differential privacy requires that the greater the hamming distance between x, x' is, the more distinguishable they are allowed to be.

This concept can be naturally extended to any set of secrets $\mathcal{X}$, equipped with a metric $d_{\mathcal{X}}$ [14, 12]. The distance $d_{\mathcal{X}}(x, x')$ expresses the *distinguishability level* between x and x': if the distance is small then the secrets should remain indistinguishable, while secrets far away from each other are allowed to be distinguished by the adversary. The metric should be chosen depending on the application at hand and the semantics of the privacy notion that we try to achieve.

Following the notation of [12], a mechanism is a probabilistic function $K : \mathcal{X} \to \mathcal{P}(\mathcal{Z})$, where $\mathcal{Z}$ is a set of *reported values* (assumed finite for the purposes of this paper). The similarity between probability distributions can be measured by the multiplicative distance $d_{\mathcal{P}}$ defined as $d_{\mathcal{P}}(\mu_1, \mu_2) = \sup_{z \in \mathcal{Z}} |\ln \frac{\mu_1(z)}{\mu_2(z)}|$ with $|\ln \frac{\mu_1(z)}{\mu_2(z)}| = 0$ if both $\mu_1(z), \mu_2(z)$ are zero and ∞ if only one of them is zero. In other words, $d_{\mathcal{P}}(\mu_1, \mu_2)$ is small iff μ_1, μ_2 assign similar probabilities to each value z.

The generalized variant of differential privacy under the metric $d_{\mathcal{X}}$, called $d_{\mathcal{X}}$-privacy, is defined as follows:

DEFINITION 2. *A mechanism $K : \mathcal{X} \to \mathcal{P}(\mathcal{Z})$ satisfies $d_{\mathcal{X}}$-privacy iff:*

$$d_{\mathcal{P}}(K(x), K(x')) \leq d_{\mathcal{X}}(x, x') \qquad \forall x, x' \in \mathcal{X}$$

or equivalently $K(x)(z) \leq e^{d_{\mathcal{X}}(x,x')} K(x')(z)$ for all $x, x' \in \mathcal{X}, z \in \mathcal{Z}$. A privacy parameter ϵ can also be introduced by scaling the metric $d_{\mathcal{X}}$ (note that $\epsilon d_{\mathcal{X}}$ is itself a metric).

Differential privacy can then be expressed as ϵd_h-privacy. Moreover, different metrics give rise to various privacy notions of interest; several examples are given in [12].

2.3 Geo-indistinguishability

In the context of location based systems the secrets $\mathcal{X}$ are locations, and we can obtain a useful notion of location privacy by naturally using the Euclidean distance d_2, scaled by a security parameter ϵ. The resulting notion of ϵd_2-privacy, called ϵ-geo-indistinguishability in [9], requires that a location obfuscation mechanism should produce similar results when applied to locations that are geographically close. This prevents the service provider from inferring the user's location with accuracy, while allowing him to get approximate information required to provide the service. Following the spirit of differential privacy, this definition is independent from the prior information of the adversary.

A characterization of geo-indistinguishability from [9] provides further intuition about this notion. The characterization compares the adversary's conclusions (a posterior distribution) to his initial knowledge (a prior distribution). Since some information is supposed to be revealed (i.e. the provider will learn that the user is somewhere around Paris), we cannot expect the two distributions to coincide. However, geo-indistinguishability implies that an *informed adversary* who already knows that the user is located within a small area N, cannot improve his initial knowledge and

locate the user with higher accuracy. More details, together with a second characterization can be found in [9].

Note that geo-indistinguishability does not guarantee a small leakage under any prior; in fact no obfuscation mechanism can ensure this while offering some utility. Consider, for instance, an adversary who knows that the user is located at some airport, but not which one. Unless the noise is huge, reporting an obfuscated location will allow the exact location to be inferred, but this is unavoidable.[1].

Considering the mechanism, [9] shows that geo-indistinguishability can be achieved by adding noise to the user's location drawn from a 2-dimensional Laplace distribution. This can be easily done in polar coordinates by selecting and angle uniformly and a radius from a Gamma distribution. If a restricted set of reported locations is allowed, then the location produced by the mechanism can be mapped back to the closest among the allowed ones.

Although the Laplace mechanism provides an easy and practical way of achieving geo-indistinguishability, independently from any user profile, its utility is not always optimal. In the next section we show that by tailoring a mechanism to a prior corresponding to a specific user profile, we can achieve better utility for that prior, while still satisfying geo-indistinguishability, i.e. a privacy guarantee independent from the prior. The evaluation results in Section 4 show that the optimal mechanism can provide substantial improvements compared to the Laplace mechanism.

3. GEO-INDISTINGUISHABLE MECHANISMS OF OPTIMAL UTILITY

As discussed in the introduction, we aim at obtaining a mechanism that optimizes the tradeoff between privacy (in terms of geo-indistinguishability) and quality loss (in terms the metric QL). Our main goal is, given a set of locations $\mathcal{X}$ with a privacy metric $d_{\mathcal{X}}$ (typically the Euclidean distance), a privacy level ϵ, a user profile π and a quality metric d_Q, to find an $\epsilon d_{\mathcal{X}}$-private mechanism such that its QL is as small as possible.

We start by describing a set of linear constraints that enforce $\epsilon d_{\mathcal{X}}$-privacy, which allows to obtain an optimal mechanism as a linear optimization problem. However, the number of constraints can be large, making the approach computationally demanding as the number of locations increases. As a consequence, we propose an approximate solution that replaces $d_{\mathcal{X}}$ with the metric induced by a spanning graph. We discuss a greedy algorithm to calculate the spanning graph and analyze its running time. We also show that, if the quality and adversary metrics coincide, then the constructed (exact or approximate) mechanisms also provide optimal privacy in terms of AdvError. Finally, we discuss some practical considerations of our approach.

3.1 Constructing an optimal mechanism

The constructed mechanism is assumed to have as both input and output a predetermined finite set of locations $\mathcal{X}$. For instance, $\mathcal{X}$ can be constructed by dividing the map in a finite number of regions (of arbitrary size and shape), and selecting in $\mathcal{X}$ a representative location for each region. We also assume a prior π over $\mathcal{X}$, representing the probability of the user being at each location at any given time.

---

[1]This example is the counterpart of the well-known Terry Gross example from [11]

Given a privacy metric $d_{\mathcal{X}}$ (typically the Euclidean distance) and a privacy parameter ϵ, the goal is to construct a $\epsilon d_{\mathcal{X}}$-private mechanism K such that the *service quality loss* with respect to a quality metric d_Q is minimum. This property is formally defined below:

DEFINITION 3. *Given a prior π, a privacy metric $d_{\mathcal{X}}$, a privacy parameter ϵ and a quality metric d_Q, a mechanism K is $\epsilon d_{\mathcal{X}}$-OptQL(π, d_Q) iff:*

1. *K is $\epsilon d_{\mathcal{X}}$-private, and*

2. *for all mechanisms K', if K' is $\epsilon d_{\mathcal{X}}$-private then* $\mathrm{QL}(K, \pi, d_Q) \leq \mathrm{QL}(K', \pi, d_Q)$

Note that $\epsilon d_{\mathcal{X}}$-OptQL optimizes QL given a privacy constraint, while q-OptPriv (Definition 1) optimizes privacy, given an QL constraint.

In order for K to be $\epsilon d_{\mathcal{X}}$-private it should satisfy the following constraints:

$$k_{xz} \leq e^{\epsilon d_{\mathcal{X}}(x,x')} k_{x'z} \qquad x, x', z \in \mathcal{X}$$

Hence, we can construct an optimal mechanism by solving a linear optimization problem, minimizing $\mathrm{QL}(K, \pi, d_Q)$ while satisfying $\epsilon d_{\mathcal{X}}$-privacy:

$$\textbf{Minimize:} \quad \sum_{x,z \in \mathcal{X}} \pi_x k_{xz} d_Q(x, z)$$

$$\textbf{Subject to:} \quad k_{xz} \leq e^{\epsilon d_{\mathcal{X}}(x,x')} k_{x'z} \qquad x, x', z \in \mathcal{X}$$

$$\sum_{z \in \mathcal{X}} k_{xz} = 1 \qquad x \in \mathcal{X}$$

$$k_{xz} \geq 0 \qquad x, z \in \mathcal{X}$$

It is easy to see that the mechanism K generated by the previous optimization problem is $\epsilon d_{\mathcal{X}}$-OptQL(π, d_Q).

3.2 A more efficient method using spanners

In the optimization problem of the previous section, the $\epsilon d_{\mathcal{X}}$-privacy definition introduces $|\mathcal{X}|^3$ constraints in the linear program. However, in order to be able to manage a large number of locations, we would like to reduce this amount to a number in the order of $O(|\mathcal{X}|^2)$. One possible way to achieve this is to use the *dual form* of the linear program (shown in the appendix). The dual program has as many constraints as the variables of the primal program (in this case $|\mathcal{X}|^2$) and one variable for each constraint in the primal program (in this case $O(|\mathcal{X}|^3)$). Since the primal linear program finds the optimal solution in a finite number of steps, it is guaranteed by the strong duality theorem that dual program will also do so. However, as shown in Section 4.3, in practice the dual program does not offer a substantial improvement with respect to the primal one (a possible explanation being that, although fewer in number, the constrains in the dual program are more complex, in the sense that each one of them involves a larger number of variables).

An alternative approach is to exploit the structure of the metric $d_{\mathcal{X}}$. So far we are not making any assumption about $d_{\mathcal{X}}$, and therefore we need to specify $|\mathcal{X}|$ constraints for each pair of locations x and x'. However, it is worth noting that if the distance $d_{\mathcal{X}}$ is induced by a weighted graph (i.e. the distance between each pair of locations is the weight of a minimum path in a graph), then we only need to consider $|\mathcal{X}|$ constraints for each pair of locations that are *adjacent*

Figure 1: (a) a division of the map of Paris into a 7×5 square grid. The set of locations $\mathcal{X}$ contains the centers of the regions. (b) A spanner of $\mathcal{X}$ with dilation $\delta = 1.08$.

in the graph. An example of this is the usual definition of differential privacy: since the adjacency relation between databases induces the Hamming distance d_h, we only need to require the differential privacy constraint for each pair of databases that are adjacent in the Hamming graph (i.e. that differ in one individual).

It might be the case, though, that the metric $d_\mathcal{X}$ is not induced by any graph (other than the complete graph), and consequently the amount of constraints remains the same. In fact, this is generally the case for the Euclidean metric. Therefore, we consider the case in which $d_\mathcal{X}$ can be *approximated* by some graph-induced metric.

If G is an undirected weighted graph, we denote with d_G the distance function induced by G, i.e. $d_G(x, x')$ denotes the weight of a minimum path between the nodes x and x' in G. Then, if the set of nodes of G is $\mathcal{X}$ and the weight of its edges is given by the metric $d_\mathcal{X}$, we can approximate $d_\mathcal{X}$ with d_G. In this case, we say that G is a spanning graph, or a spanner [15, 16], of $\mathcal{X}$.

DEFINITION 4 (SPANNER). *A weighted graph $G = (\mathcal{X}, E)$, with $E \subseteq \mathcal{X} \times \mathcal{X}$ and weight function $w : E \to \mathbb{R}$ is a* spanner *of $\mathcal{X}$ if*

$$w(x, x') = d_\mathcal{X}(x, x') \quad \forall (x, x') \in E$$

Note that if G is a spanner of $\mathcal{X}$, then

$$d_G(x, x') \geq d_\mathcal{X}(x, x') \quad \forall x, x' \in \mathcal{X}$$

A main concept in the theory of spanners is that of dilation, also known as stretch factor:

DEFINITION 5 (DILATION). *Let $G = (\mathcal{X}, E)$ be a spanner of $\mathcal{X}$. The* dilation *of G is calculated as:*

$$\delta = \max_{x \neq x' \in \mathcal{X}} \frac{d_G(x, x')}{d_\mathcal{X}(x, x')}$$

A spanner of $\mathcal{X}$ with dilation δ is called a δ-spanner of $\mathcal{X}$.

Informally, a δ-spanner of $\mathcal{X}$ can be considered an approximation of the metric $d_\mathcal{X}$ in which distances between nodes are "stretched" by a factor of at most δ. Spanners are generally used to approximate distances in a geographic network without considering the individual distances between each pair of nodes. An example of a spanner for a grid in the map can be seen in Figure 1.

If G is a δ-spanner of $\mathcal{X}$, then it holds that

$$d_G(x, x') \leq \delta d_\mathcal{X}(x, x') \quad \forall x, x' \in \mathcal{X}$$

which leads to the following proposition:

PROPOSITION 1. *Let $\mathcal{X}$ be a set of locations with metric $d_\mathcal{X}$, and let G be a δ-spanner of $\mathcal{X}$. If a mechanism K for $\mathcal{X}$ is $\frac{\epsilon}{\delta} d_G$-private, then K is $\epsilon d_\mathcal{X}$-private.*

We can then propose a new optimization problem to obtain a $\epsilon d_\mathcal{X}$-private mechanism. If $G = (\mathcal{X}, E)$ is a δ-spanner of $\mathcal{X}$, we require not the constraints corresponding to $\epsilon d_\mathcal{X}$-privacy, but those corresponding to $\frac{\epsilon}{\delta} d_G$-privacy instead, that is, $|\mathcal{X}|$ constraints for each edge of G:

Minimize: $\quad \sum_{x,z \in \mathcal{X}} \pi_x k_{xz} d_Q(x, z)$

Subject to: $\quad k_{xz} \leq e^{\frac{\epsilon}{\delta} d_G(x,x')} k_{x'z} \quad z \in \mathcal{X}, (x, x') \in E$

$$\sum_{x \in \mathcal{X}} k_{xz} = 1 \qquad\qquad x \in \mathcal{X}$$

$$k_{xz} \geq 0 \qquad\qquad x, z \in \mathcal{X}$$

Since the resulting mechanism is $\frac{\epsilon}{\delta} d_G$-private, by Proposition 1 it must also be $\epsilon d_\mathcal{X}$-private. However, the number of constraints in induced by $\frac{\epsilon}{\delta} d_G$-privacy is now $|E||\mathcal{X}|$. Moreover, as discussed in the next section, for any $\delta > 1$ there is an algorithm that generates a δ-spanner with $O(\frac{|\mathcal{X}|}{\delta-1})$ edges, which means that, fixing δ, the total number of constraints of the linear program is $O(|\mathcal{X}|^2)$.

It is worth noting that although $\epsilon d_\mathcal{X}$-privacy is guaranteed, optimality is lost: the obtained mechanism is $\frac{\epsilon}{\delta} d_G$-OPTQL($\pi, d_Q$) but not necessarily $\epsilon d_\mathcal{X}$-OPTQL(π, d_Q), since the set of $\frac{\epsilon}{\delta} d_G$-private mechanisms is a subset of the set of $\epsilon d_\mathcal{X}$-private mechanisms. The QL of the obtained mechanism will now depend on the dilation δ of the spanner: the smaller δ is, the closer the QL of the mechanism will be from the optimal one. However, if δ is too small then the number of edges of the spanner will be large, and therefore the number of constraints in the linear program will increase. In fact, when $\delta = 1$ the mechanism obtained is also $\epsilon d_\mathcal{X}$-OPTQL(π, d_Q) (since d_G and $d_\mathcal{X}$ coincide), but the amount of constraints is in general $O(|\mathcal{X}|^3)$. In consequence, there is a tradeoff between the accuracy of the approximation and the number of constraints in linear program.

3.3 An algorithm to construct a δ-spanner

The previous approach requires to compute a spanner for $\mathcal{X}$. Moreover, given a dilation factor δ, we are interested in generating a δ-spanner with a reasonably small number of edges. In this section we describe a simple greedy algorithm to get a δ-spanner of $\mathcal{X}$, presented in [15]. This procedure (described in Algorithm 1) is a generalization of Kruskal's minimum spanning tree algorithm.

The idea of the algorithm is the following: we start with a spanner with an empty set of edges (lines 2-3). In the main loop we consider all possible edges (that is, all pairs of locations) in *increasing order* with respect to the distance function $d_\mathcal{X}$ (lines 4-8), and if the weight of a minimum path between the two corresponding locations in the current graph is bigger than δ times the distance between them, we add the edge to the spanner. By construction, at the end of the procedure, graph G is a δ-spanner of $\mathcal{X}$.

A crucial result presented in [15] is that, in the case where $\mathcal{X}$ is a set of points in the Euclidean plane, the degree of each node in the generated spanner only depends on the dilation factor:

Algorithm 1 Algorithm to get a δ-spanner of $\mathcal{X}$

1: **procedure** GETSPANNER($\mathcal{X}, d_{\mathcal{X}}, \delta$)
2:     $E := \emptyset$
3:     $G := (\mathcal{X}, E)$
4:     **for all** $(x, x') \in (\mathcal{X} \times \mathcal{X})$ **do**   ▷ taken in increasing order wrt $d_{\mathcal{X}}$
5:         **if** $d_G(x, x') > \delta d_{\mathcal{X}}(x, x')$ **then**
6:             $E := E \cup \{(x, x')\}$
7:         **end if**
8:     **end for**
9:     **return** G
10: **end procedure**

THEOREM 1. *Let $\delta > 1$. If G is a δ-spanner for $\mathcal{X} \subseteq \mathbb{R}^2$, with the Euclidean distance d_2 as metric, then the degree of each node in the spanner constructed by Algorithm 1 is $O(\frac{1}{\delta-1})$.*

This result is useful to estimate the total number of edges in the spanner, since our goal is to generate a *sparse* spanner, i.e. a spanner with $O(|\mathcal{X}|)$ edges.

Considering the running time of the algorithm, since the main loop requires all pair of regions to be sorted increasingly by distance, we need to perform this sorting before the loop. This step takes $O(|\mathcal{X}|^2 \log |\mathcal{X}|)$. The main loop performs a minimum-path calculation in each step, with $|\mathcal{X}|^2$ total steps. If we use, for instance, Dijkstra's algorithm, each of these operations can be done in $O(|E| + |\mathcal{X}| \log |\mathcal{X}|)$. If we select δ so that the final amount of edges in the spanner is linear, i.e. $|E| = O(|\mathcal{X}|)$, we can conclude that the total running time of the main loop is $O(|\mathcal{X}|^3 \log |\mathcal{X}|)$. This turns out to be also the complexity of the whole algorithm.

A common problem in the theory of spanners is the following: given a set of points $\mathcal{X} \subseteq \mathbb{R}^2$ and a maximum amount of edges m, the goal is to find the spanner with *minimum* dilation with at most m edges. This has been proven to be NP-Hard ([17]). In our case, we are interested in the analog of this problem: given a maximum tolerable dilation factor δ, we want to find a δ-spanner with minimum amount of edges. However, we can see that the first problem can be expressed in terms of the second (for instance, with a binary search on the dilation factor), which means that the second problems must be at least NP-Hard as well.

3.4 ADVERROR **of the obtained mechanism**

As discussed in 2.1, the privacy of a location obfuscation mechanism can be expressed in terms of ADVERROR for an adversary metric d_A. In [8], the problem of optimizing privacy for a given QL constraint is studied, providing a method to obtain a q-OPTPRIV(π, d_A, d_Q) mechanism for any q, π, d_Q, d_A.

In our case, we optimize QL for a given privacy constraint, constructing a $\epsilon d_{\mathcal{X}}$-OPTQL(π, d_Q) mechanism. We now show that, if d_Q and d_A coincide, the mechanism generated by any of the two optimization problems of the previous sections is also q-OPTPRIV(π, d_Q, d_Q).

ADVERROR corresponds to an adversary's remapping H that minimizes his expected error with respect to the metric d_A and his prior knowledge π. A crucial observation is that $d_{\mathcal{X}}$-privacy is closed under remapping.

LEMMA 1. *Let K be a $d_{\mathcal{X}}$-private mechanism, and let H be a remapping. Then KH is $d_{\mathcal{X}}$-private.*

Now let K be a $d_{\mathcal{X}}$-OPTQL(π, d_Q) mechanism and H a remapping. Since KH is $d_{\mathcal{X}}$-private (Lemma 1) and K is optimal among all such mechanisms, we have that:

$$\mathrm{QL}(K, \pi, d_Q) \leq \mathrm{QL}(KH, \pi, d_Q) \quad \forall H$$

As a consequence, assuming that d_Q and d_A coincide, the adversary minimizes his expected error by applying no remapping at all (i.e. the identity remapping), which means that ADVERROR(K, π, d_Q) = QL(K, π, d_Q) and therefore K must be q-OPTPRIV(π, d_Q, d_Q).

THEOREM 2. *If a mechanism K is $d_{\mathcal{X}}$-OPTQL(π, d_Q) then it is also q-OPTPRIV(π, d_Q, d_Q) for $q = $ QL(K, π, d_Q).*

It is important to note that Theorem 2 holds for any metric $d_{\mathcal{X}}$. This means that both mechanisms obtained as result of the optimization problems presented in Sections 3.1 and 3.2 are q-OPTPRIV(π, d_Q, d_Q) – since they are $\epsilon d_{\mathcal{X}}$-OPTQL(π, d_Q) and $\frac{\epsilon}{\delta} d_G$-OPTQL($\pi, d_Q$) respectively – however for a different value of q. In fact, in contrast to the method of [8] in which the quality bound q is given as a parameter, our method optimizes the QL given a privacy bound. Hence, the resulting mechanism will be q-OPTPRIV(π, d_Q, d_Q), but for a q that is not known in advance and will depend on the privacy constraint ϵ and the dilation factor δ. The greater the ϵ is (i.e. the higher the privacy), or the lower the δ is (i.e. the better the approximation), the lower the quality loss q of the obtained mechanism will be.

Finally, we must remark that this result only holds in the case where the metrics d_Q, d_A coincide. If the metrics differ, e.g. the quality is measured in terms of the Euclidean distance (the user is interested in accuracy) but the adversary uses the binary distance (he is only interested in the exact location), then this property will no longer be true.

3.5 Practical considerations

We conclude this section with a discussion on the practical applicability of location obfuscation. First, it should be noted that, although constructing an optimal mechanism is computationally demanding, once the matrix K is computed, obfuscating a location x only involves drawing a reported location from the distribution $K(x)$ which is computationally trivial. Moreover, although obfuscation is meant to happen on the user's smartphone, computing the mechanism can be offloaded to an external server and even parallelized. The user only needs to transmit $\pi, \epsilon d_{\mathcal{X}}, d_Q$ (which are considered public) and receive K, and the computation only needs to be performed occasionally, to adapt to changes in the user profile.

Second, an important feature of obfuscation mechanisms is that they require no cooperation from the service provider, who simply receives a location and has no way of knowing whether it is real or not. Obfuscation can happen on the user's device, at the operating system or browser level, which is crucial since the user has strong incentives to apply it while the service provider does not. The user's device could also perform filtering of the results, as described in [9].

Finally, we argue that the common idea that users of LBSs are willing to give up their privacy is misleading: the only alternative offered is not to use the service. The usage of browser extensions such as "Location Guard" [18] shows that users do care about their privacy and that obfuscation can be

(a)   (b)

Figure 2: (a) Division of the map of Beijing into regions of size 0.658 x 0.712 km. The density of each region represents its "score", that is, how frequently users visit it. (b) The 50 selected regions. These regions are the ones with highest density between the whole set of regions.

Figure 3: Boxplot of the location privacy provided by the three different mechanisms under considered priors. The OptQL mechanism was constructed with $\epsilon = 1.07$ and $\delta = 1.05$.

a practical approach for using existing services in a privacy friendly way.

4. EVALUATION

In this section we evaluate the technique for constructing optimal mechanisms described in the previous sections. We perform two kinds of evaluation: first, a comparison with other mechanisms, namely the one of Shokri et al. and the Planar Laplace mechanism. Second, a performance evaluation of the spanner approximation technique.

The comparison with other mechanisms is performed with respect to both privacy and quality loss. For privacy, the main motivation is to evaluate the mechanisms' privacy under different priors, and in particular under priors different than the one they were constructed with. Following the motivating scenario of the introduction, we consider that a user's profile can vary substantially between different time periods of the day, and simply by taking into account the time of a query, the adversary can obtain a much more informative prior which leads to a lower privacy. For the purposes of the evaluation, we consider priors corresponding to four different time periods: the full day, the morning (7am to noon), afternoon (noon to 7pm) and night (7pm to 7am). Then we construct the mechanisms using the full day prior and compare their privacy for all time periods.

We perform our evaluation on two widely used datasets: GeoLife [19, 20, 21] and T-Drive [22, 23]. The results of GeoLife are presented in detail in the following sections, while, due to space restrictions, those of T-Drive (which are in general similar) are summarized in Section 4.4.

4.1 The GeoLife dataset

The GeoLife GPS Trajectories dataset contains 17621 traces from 182 users, moving mainly in the north-west of Beijing, China, in a period of over five years (from April 2007 to August 2012). The traces show users performing routinary tasks (like going to and from work), and also traveling, shopping, and doing other kinds of entertainment or unusual activities. Besides, the traces were logged by users using different means of transportation, like walking, public transport or bike. More than 90% of the traces were logged in a dense representation, meaning that the individual points in the trace were reported every 1-5 seconds or every 5-10 meters. Since user behaviour changes over time, and the mechanism

should be occasionally reconstructed, we restrict each user's traces to a 90 days period, and in particular to the one with the greatest number of recorded traces, so that the prior is as informative as possible.

4.2 Mechanism comparison wrt privacy and quality loss

For the evaluation, we divide the map of Beijing into a grid of regions 0.658 km wide and 0.712 km high, displayed in Figure 2a. To avoid users for which little information is available, we only keep those having at least 20 recorded points within the grid area for each one of the time periods. Whenever we count points, those falling within the same grid region during the same hour are counted only once, to prevent traces with a huge number of points in the same region (e.g. the user's home) from completely skewing the results. After this filtering, we end up with 116 users (64% of the total 182).

We then proceed to calculate the 50 "most popular" regions of the grid as follows: for each user, we select the 30 regions in which he spends the greatest amount of time. A region's "score" is the number of users that have it in their 30 highest ranked ones. Then we select the 50 regions with the highest score.

Figure 2a shows the division of the map into regions, with the opacity representing the score of each of them, while Figure 2b shows the 50 regions with highest score. We can see that most of the selected regions are located in the southeast of the Haidian district, and all of them are located in the north-west of Beijing. We consider the set of locations $\mathcal{X}$ to be the centers of the selected regions, and the metric $d_{\mathcal{X}}$ to be the Euclidean distance between these centers, i.e. $d_{\mathcal{X}} = d_2$.

Finally, a second filtering is performed, again keeping users with at least 20 points in each time period, but this time considering only the 50 selected regions. After this, we end up with a final set of 86 users (46% of the total 182).

In this section, we evaluate the location privacy and the utility of three different mechanisms under the several prior distributions for each user. These priors correspond to different parts of the day (all day, morning, afternoon and night), and are computed by counting the number of points, logged in the corresponding time period, that fall in each of the se-

Figure 4: Quality loss of the OptQL and PL mechanisms for different values of ϵ. The mechanisms were calculated for all users. Here, points represent the utility for every user, while the two lines join the medians for each mechanism and each value of ϵ.

lected regions (again, counting only once those points logged within the same hour), and then by normalizing these numbers to obtain a probability distribution.

We start by evaluating the location privacy provided by the different mechanisms. However, we must note that in general location privacy mechanisms do not satisfy $\epsilon d_{\mathcal{X}}$-privacy unless they are specifically designed to do so. Therefore, for this evaluation, we measure location privacy with the metric ADVERROR, proposed in [8] and described in Section 2.1, which measures the expected error of the attacker under a given prior distribution. In order to perform a fair comparison, we construct the mechanisms in such a way that their QL coincide. The first step is to select a privacy level ϵ and a dilation δ, and then to construct the mechanism described in Section 3.2. We will call this mechanism OptQL. This mechanism has a QL of $q = \text{QL}(\text{OptQL}, \pi, d_2)$. We then continue by constructing the optimal mechanism of Shokri et al [8], and setting the QL as q. We call this mechanism OptPriv. Finally, we compute a discretized version of the Planar Laplace mechanism of Andrés et al [9]. under a privacy constraint ϵ' (in general different from ϵ) such that the QL of this mechanism is also q. We call this mechanism PL. Note that at the end of this process, by construction, the QL of the three mechanisms is q.

We begin the evaluation comparing the location privacy of each mechanism for each of the selected users, under the four constructed priors. We fix $\epsilon = 1.07$ (which intuitively corresponds to a ratio of 2 between the probability for two regions adjacent in the grid to report the same observed location) and $\delta = 1.05$. Figure 3 shows a boxplot of the location privacy (in km) offered by the different mechanisms under each prior. In all four cases, the general performance of our mechanism is better than that of the others, with the only exception being the all-day prior (which is the one used in the construction of the mechanisms) since, as explained in

Section 3.4, OptQL and OptPriv are q-OptPriv(π, d_2, d_2) and therefore offer the same privacy.

Finally, to show the benefits of using a mechanism with optimal utility, we compare now the QL of the mechanisms OptQL and PL when both mechanisms are generated with the same privacy level ϵ. We can see the results in Figure 4. The OptQL mechanism clearly offers a better utility to the user, while guaranteeing the same level of geo-indistinguishability.

4.3 Performance of the approximation algorithm

We recall from Section 3.2 that if we consider a large number of locations in $\mathcal{X}$, then the number of constraints in the linear program might be large. Hence, we introduced a method based on a spanning graph G to reduce the total number of constraints of the linear program. However, in general the obtained mechanism is no longer $\epsilon d_{\mathcal{X}}$-OptQL(π, d_Q), and therefore it has a higher QL than the optimal one.

In this section we study the tradeoff between the increase in the QL of the mechanism and the reduction in the number of constraints of the optimization problem, as a consequence of using our approximation technique. We also show how this reduction affects the running time of the whole approach. We start by constructing the OptQL mechanism for all selected users and for different dilations in the range from 1.05 to 2.0, in all cases considering $\epsilon = 1.07$ as before. We then measure the QL of each mechanism under the user profile. We can see the results in Figure 5a. It is clear that the QL increases slowly with respect to the dilation: the median value is 0.946 km for $\delta = 1.05$, is 0.972 km for $\delta = 1.1$, and 1.018 km for $\delta = 1.2$. Therefore we can deduce that, for a reasonable approximation, the increase in the quality loss is not really significant. It is worth noting that we do not show the QL for $\delta = 1$ in the plot (corresponding to the case where $d_{\mathcal{X}}$ and d_G are the same). The reason is that in that case the number of constraints is really high, and therefore it takes a lot of time to generate one instance of the mechanism (and much more time to generate it for the 86 users considered).

The relation between the dilation and the number of constraints is shown in Figure 5b. Note that this number is independent from the user, and therefore it is enough to calculate it for just one of them. It is clear that the number of constraints decreases exponentially with respect to the dilation, and therefore even for small dilations (which in turn mean good approximations) the number of constraints is significantly reduced with the proposed approximation technique. For instance, we have 87250 constraints for $\delta = 1$ (the optimal case), and 25551 constraints for $\delta = 1.05$. This represents a decrease of 71% with respect to the optimal case, with only 1.05 approximation ratio.

It is also worth noting that, between $\delta = 1.4$ and $\delta = 1.45$ there is a pronounced decrease in the number of constraints (Figure 5b) and *also* a decrease in the QL (Figure 5a). This might seem counterintuitive at first, since one would expect that a worse approximation should always imply a higher loss of quality. However, there is a simple explanation: although the spanner with $\delta = 1.45$ has a higher worst-case approximation ratio, the average-case ratio is actually better that the one of the spanner with $\delta = 1.4$. This phenomenon

Figure 5: (a) **Boxplot of the relation between QL and dilation for the mechanism OptQL with privacy constraint $\epsilon = 1.07$. The spanner is calculated with the greedy algorithm presented in Section 3.3. (b) Relation between the approximation ratio and the number of constraints in the linear program. This number is independent from the user and form the value of ϵ.**

is a consequence of the particular topology of the set of locations and to the algorithm used to get the spanner.

Finally, we measure the running time of the method used to generate the OptQL mechanism, under different methods to solve the linear optimization problem. The experiments were performed in a 2.8 GHz Intel Core i7 MacBook Pro with 8 GB of RAM running Mac OS X 10.9.1, and the source code for the method was written in C++, using the routines in the GLPK library for the linear program. We compare the performance of three different methods included in the library: the simplex method in both its primal and dual form, and the primal-dual interior-point method. Besides, we run these methods on both the primal linear program presented in Section 3.2 and its dual form, presented in Appendix B. Since the running time depends mainly on the number of locations being considered, in the experiments we focus on just one user of the dataset, and we fix the privacy level as $\epsilon = 1.07$. The results can be seen in Table 1. Some fields are marked with "1h+", meaning that the execution took more than one hour, after which it was stopped. Others are marked with "Error", meaning that the execution stopped before one hour with an error[2]. A particular case of error happened when running the interior-point method on the dual linear program, where all executions ended with a "numerical instability" error (and therefore this case is not included in the table). From the results we can observe that:

- The only two methods that behave consistently (that never finish with error, and the running time increases when the dilation decreases) are the dual simplex and the interior-point methods, both when applied to the primal program.

- From these, the interior-point method performs better in the case of bigger dilation, while it does it much worse for very small ones.

- Somewhat surprisingly, the dual linear program does not offer a significant performance improvement, specially when compared with the interior-point method.

---

[2]The actual error message in this case was: "Error: unable to factorize the basis matrix (1). Sorry, basis recovery procedure not implemented yet"

		Primal simplex		Dual simplex		Interior
$|\mathcal{X}|$	δ	Pr. LP	Du. LP	Pr. LP	Du. LP	Pr. LP
	1.0	57s	1h+	40s	45s	49m 20s
	1.1	46.4s	5.2	5.9s	15.5s	7.5s
50	1.2	4m 37s	2s	4s	1h+	2.7s
	1.5	2s	1s	2s	3s	0.5s
	2.0	Error	1s	2s	2s	0.5s
	1.0	1h+	1h+	29m 26s	1h+	1h+
	1.1	1h+	Error	1m 12s	2m 19s	55s
75	1.2	1h+	Error	42s	48.4s	11.7s
	1.5	1h+	5m 55s	19.2s	1h+	2.2s
	2.0	1h+	21.8s	27.2s	15.5s	1.7s

Table 1: Execution times of our approach for 50 and 75 locations, for different values of δ, and using different methods to solve the linear program.

In the case of OptPriv, the mechanism is generated using Matlab's linear program solver (source code kindly provided by the authors of [8]). We generated the mechanism for the same cases, and observed that the running time mainly depends on the number of regions: for 50 regions, the mechanism is generated in approximately 1 minute, while for 75 regions it takes about 11 minutes.

4.4 The T-Drive dataset

In order to reaffirm the validity of the proposed approach, we performed the same evaluation in a different dataset: the T-Drive trajectories dataset. This dataset contains traces of 10357 taxis in Beijing, China, during the period of one week. The total distance of the traces in this dataset is about 9 million kilometres, with more than 15 million reported points. The average time between consecutive points in a trace is 177 seconds, and the average distance is 623 meters.

Due to the huge amount of users in this dataset, we started the evaluation process by blindly selecting (using a standard random function) 5% of the total number users (about 532 users out of 10357). We then perform the same steps as described in the previous sections, particularly those described in Section 4.2. In Figure 6 we can see the comparison of the location privacy for the different mechanisms. We can see that, also for this dataset, the privacy level of OptQL is, in general, as good as the one of OptPriv, and always better than the one of PL. In particular, the median value for

Figure 6: Boxplot of the location privacy for the T-Drive dataset. The median value of the location privacy for OptQL is always as good as the one of the other mechanisms.

OPTQL is always higher than the corresponding one for the other mechanisms (again, with the exception of the all day prior, for which we know that these values coincide). We can also see in Figure 7 the comparison in terms of utility of the mechanisms OPTQL and PL. Again, the quality loss of OPTQL is, in all cases, better than the one of PL. This is to be expected, since, from all mechanisms providing a certain geo-indistinguishability, OPTQL is the one with optimal utility (or really close to the optimal utility when the approximation is used).

5. CONCLUSION AND RELATED WORK

Related work

In the last years, a large number of location-privacy protection techniques, diverse both in nature and goals, have been

Figure 7: Quality loss of the OptQL and PL mechanisms for different values of ϵ, using the data in the T-Drive dataset. The loss of quality of OptQL is always smaller than the one of PL.

proposed and studied. Many of these aim at allowing the user of an LBS to hide his *identity* from the service provider. Several approaches are based in the notion of k-anonymity [24, 25, 26], requiring that the attacker cannot identify a user from at least other $k-1$ different users. Others are based on the idea of letting the users use pseudonyms to interact with the system, and on having regions (*mix zones*, [4, 6]), where the users can change their pseudonyms without being traced by the system. All these approaches are incomparable with ours, since ours aims at hiding the *location* of the user and not his identity.

Many approaches to location privacy are based on obfuscating the position of the user. A common technique for this purpose is *cloaking* [27, 28, 29, 25], which consists in blurring the user's location by reporting a region to the service provider. Another technique is based on adding *dummy locations*[30, 31, 5] to the request sent to the service provider. In order to preserve privacy, these dummy locations should be generated in such a way that they look equally likely to be the user's real position. A different approach is to construct mechanisms that provide optimal privacy under certain quality constraints [8] (an approach dual to ours, as discussed in the introduction), while [32] additionally takes into account bandwidth constraints. Finally, collaborative models have been proposed [33], where privacy is achieved with a peer-to-peer scheme where users avoid querying the service provider whenever they can find the requested information among their peers.

Differential Privacy has also been used in the context of location privacy; however, it is in general used to protect *aggregate* location information. For instance, [34] presents a way to statistically simulate the location data from a database while providing privacy guarantees. In [35], a quad tree spatial decomposition technique is used to achieve differential privacy in a database with location patter mining capabilities. On the other hand, Dewri [36] proposes a combination of differential privacy and k-anonymity for the purposes of hiding the location of a single individual. The proposed definition requires that the distances between the probability distributions corresponding to k fixed locations (defined as the anonymity set) should not be greater than the privacy parameter ϵ.

The work closest to ours is [37], which independently proposes a linear programming technique to construct optimal obfuscation mechanisms wrt either ADVERROR or geo-indistinguishability. Although there is an overlap in the main construction (the optimization problem of Section 3.1), most of the results are substantially different. The approximation technique of [37] consists of discarding some of the geo-indistinguishability constraints when the distance involved is larger than a certain lower bound. This affects the geo-indistinguishability guarantees of the mechanism, although the effect can be tuned by properly selecting the bound for discarding constraints. On the other hand, our approximation technique, based on spanning graphs, can be used to reduce the number of constraints from cubic to quadratic without jeopardizing the privacy guarantees, by accepting a small decrease on the utility. Moreover, we show that the mechanism obtained from this optimization problem is also optimal wrt ADVERROR (Theorem 2), which is an important property of the proposed method. Finally, the evaluation methods are substantially different: in [37] the employed set of prior distributions differ in their level of

entropy (priors with low entropy are considered more informative). In our work, we obtain the different priors by combining the distribution of the user (assumed to be known by the adversary) with some public available information (for instance, the time of the day).

Finally, $d_\mathcal{X}$-privacy has been used in [38] to capture *fairness*, instead of privacy. The goal is to construct a fair mechanism that produces similar reported values for "similar" users, the similarity being captured by the metric. As in our work, the construction involves solving an optimization problem, however no technique is used to reduce the number of constraints.

Conclusion

In this paper we have developed a method to generate a mechanism for location privacy that combines the advantages of the geo-indistinguishability privacy guarantee of [9] and the optimal mechanism of [8]. Since linear optimization is computationally demanding, we have provided a technique to reduce the total number of constraints in the linear program, based on the use of a spanning graph to approximate distances between locations, which allows a huge reduction on the number of constraints with only a small decrease in the utility. Finally, we have evaluated the proposed approach using traces from real users, and we have compared both the privacy and the running time of our mechanism with that of [8]. It turns out that our mechanism offers better privacy guarantees when the side knowledge of the attacker is different from the distribution used to construct the mechanisms. Besides, for a reasonably good approximation factor, we have showed that our approach performs much better in terms of running time.

6. ACKNOWLEDGEMENTS

7. REFERENCES

[1] Freudiger, J., Shokri, R., Hubaux, J.P.: Evaluating the privacy risk of location-based services. In: Proc. of FC'11. Volume 7035 of LNCS., Springer (2011) 31–46

[2] Golle, P., Partridge, K.: On the anonymity of home/work location pairs. In: Proc. of PerCom'09. Volume 5538 of LNCS. Springer-Verlag (2009) 390–397

[3] Krumm, J.: Inference attacks on location tracks. In: Proc. of PERVASIVE. Volume 4480 of LNCS., Springer (2007) 127–143

[4] Beresford, A.R., Stajano, F.: Location privacy in pervasive computing. IEEE Pervasive Computing **2**(1) (2003) 46–55

[5] Chow, R., Golle, P.: Faking contextual data for fun, profit, and privacy. In: Proc. of WPES, ACM (2009) 105–108

[6] Freudiger, J., Shokri, R., Hubaux, J.P.: On the optimal placement of mix zones. In: Proc. of PETS 2009. Volume 5672 of LNCS., Springer (2009) 216–234

[7] Hoh, B., Gruteser, M., Xiong, H., Alrabady, A.: Preserving privacy in gps traces via uncertainty-aware path cloaking. In: Proc. of CCS, ACM (2007) 161–171

[8] Shokri, R., Theodorakopoulos, G., Troncoso, C., Hubaux, J.P., Boudec, J.Y.L.: Protecting location privacy: optimal strategy against localization attacks. In: Proc. of CCS, ACM (2012) 617–627

[9] Andrés, M.E., Bordenabe, N.E., Chatzikokolakis, K., Palamidessi, C.: Geo-indistinguishability: differential privacy for location-based systems. In: Proc. of CCS, ACM (2013) 901–914

[10] Shokri, R., Theodorakopoulos, G., Boudec, J.Y.L., Hubaux, J.P.: Quantifying location privacy. In: Proc. of S&P, IEEE (2011) 247–262

[11] Dwork, C., Mcsherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Proc. of TCC. Volume 3876 of LNCS., Springer (2006) 265–284

[12] Chatzikokolakis, K., Andrés, M.E., Bordenabe, N.E., Palamidessi, C.: Broadening the scope of Differential Privacy using metrics. In: Proc. of PETS. Volume 7981 of LNCS., Springer (2013) 82–102

[13] Chatzikokolakis, K., Palamidessi, C., Stronati, M.: A predictive differentially-private mechanism for mobility traces. In: Proc. of PETS. Volume 8555 of LNCS., Springer (2014) 21–41

[14] Reed, J., Pierce, B.C.: Distance makes the types grow stronger: a calculus for differential privacy. In: Proc. of ICFP, ACM (2010) 157–168

[15] Narasimhan, G., Smid, M.: Geometric spanner networks. CUP (2007)

[16] Sack, J., Urrutia, J.: Handbook of Computational Geometry. Elsevier Science (1999)

[17] Klein, R., Kutz, M.: Computing Geometric Minimum-Dilation Graphs is NP-Hard. In: Proc. of the GD. Volume 4372., Springer (2006) 196–207

[18] : Location Guard. https://github.com/chatziko/location-guard.

[19] Zheng, Y., Li, Q., Chen, Y., Xie, X., Ma, W.Y.: Understanding Mobility Based on GPS Data. In: Proc. of UbiComp 2008. (2008)

[20] Zheng, Y., Zhang, L., Xie, X., Ma, W.Y.: Mining interesting locations and travel sequences from GPS trajectories. In: Proc. of WWW 2009. (2009)

[21] Zheng, Y., Xie, X., Ma, W.Y.: Geolife: A collaborative social networking service among user, location and trajectory. IEEE Data Eng. Bull. **33**(2) (2010) 32–39

[22] Yuan, J., Zheng, Y., Xie, X., Sun, G.: Driving with knowledge from the physical world. In: The 17th ACM SIGKDD international conference on Knowledge Discovery and Data mining, KDD '11. (2011)

[23] Yuan, J., Zheng, Y., Zhang, C., Xie, W., Xie, X., Sun, G., Huang, Y.: T-drive: driving directions based on taxi trajectories. In: GIS. (2010) 99–108

[24] Gruteser, M., Grunwald, D.: Anonymous usage of location-based services through spatial and temporal cloaking. In: Proc. of MobiSys, USENIX (2003)

[25] Gedik, B., Liu, L.: Location privacy in mobile systems: A personalized anonymization model. In: Proc. of ICDCS, IEEE (2005) 620–629

[26] Mokbel, M.F., Chow, C.Y., Aref, W.G.: The new casper: Query processing for location services without compromising privacy. In: Proc. of VLDB, ACM (2006) 763–774

[27] Bamba, B., Liu, L., Pesti, P., Wang, T.: Supporting anonymous location queries in mobile environments with privacygrid. In: Proc. of WWW, ACM (2008) 237–246

[28] Duckham, M., Kulik, L.: A formal model of obfuscation and negotiation for location privacy. In: Proc. of PERVASIVE. Volume 3468 of LNCS., Springer (2005) 152–170

[29] Xue, M., Kalnis, P., Pung, H.: Location diversity: Enhanced privacy protection in location based services. In: Proc. of LoCA. Volume 5561 of LNCS., Springer (2009) 70–87

[30] Kido, H., Yanagisawa, Y., Satoh, T.: Protection of location privacy using dummies for location-based services. In: Proc. of ICDE Workshops. (2005) 1248

[31] Shankar, P., Ganapathy, V., Iftode, L.: Privately querying location-based services with SybilQuery. In: Proc. of UbiComp, ACM (2009) 31–40

[32] Herrmann, M., Troncoso, C., Diaz, C., Preneel, B.: Optimal sporadic location privacy preserving systems in presence of bandwidth constraints. In: Proc. of WPES. (2013)

[33] Shokri, R., Theodorakopoulos, G., Papadimitratos, P., Kazemi, E., Hubaux, J.P.: Hiding in the mobile crowd: Location privacy through collaboration. In: Proc. of the TDSC, IEEE (2014)

[34] Machanavajjhala, A., Kifer, D., Abowd, J.M., Gehrke, J., Vilhuber, L.: Privacy: Theory meets practice on the map. In: Proc. of ICDE, IEEE (2008) 277–286

[35] Ho, S.S., Ruan, S.: Differential privacy for location pattern mining. In: Proc. of SPRINGL, ACM (2011) 17–24

[36] Dewri, R.: Local differential perturbations: Location privacy under approximate knowledge attackers. IEEE Trans. on Mobile Computing **99**(PrePrints) (2012) 1

[37] Shokri, R.: Optimal user-centric data obfuscation. Technical report, ETH Zurich (2014) `http://arxiv.org/abs/1402.3426`.

[38] Dwork, C., Hardt, M., Pitassi, T., Reingold, O., Zemel, R.S.: Fairness through awareness. In: Proc. of ITCS, ACM (2012) 214–226

APPENDIX

A. PROOFS

PROPOSITION 1. *Let $\mathcal{X}$ be a set of locations with metric $d_{\mathcal{X}}$, and let G be a δ-spanner of $\mathcal{X}$. If a mechanism K for $\mathcal{X}$ is $\frac{\epsilon}{\delta}d_G$-private, then K is $\epsilon d_{\mathcal{X}}$-private.*

PROOF. This proposition is a direct consequence of the property

$$d_G(x,x') \leq \delta d_{\mathcal{X}}(x,x') \quad \forall x,x' \in \mathcal{X}$$

and one of the results presented in [12], which states that if two metrics $d_{\mathcal{X}}$ and $d_{\mathcal{Y}}$ are such that $d_{\mathcal{X}} \leq d_{\mathcal{Y}}$ (point-wise), then $d_{\mathcal{X}}$-privacy implies $d_{\mathcal{Y}}$-privacy. □

LEMMA 1. *Let K be a $d_{\mathcal{X}}$-private mechanism, and let H be a remapping. Then KH is $d_{\mathcal{X}}$-private.*

PROOF. We know that

$$(KH)_{x\hat{x}} = \sum_{z \in \mathcal{X}} k_{xz} h_{z\hat{x}}, \quad \forall x, \hat{x} \in \mathcal{X}$$

Since K is $\epsilon d_{\mathcal{X}}$-private, we also know that

$$k_{xz} \leq e^{\epsilon d_{\mathcal{X}}(x,x')} k_{x'z}, \quad \forall x,x',z \in \mathcal{X}$$

Therefore, given $x, x' \in \mathcal{X}$, it holds that for all $\hat{x} \in \mathcal{X}$:

$$\begin{aligned}
(KH)_{x\hat{x}} &= \sum_{z \in \mathcal{X}} k_{xz} h_{z\hat{x}} \\
&\leq \sum_{z \in \mathcal{X}} e^{\epsilon d_{\mathcal{X}}(x,x')} k_{x'z} h_{z\hat{x}} \\
&= e^{\epsilon d_{\mathcal{X}}(x,x')} \sum_{z \in \mathcal{X}} k_{x'z} h_{z\hat{x}} \\
&= e^{\epsilon d_{\mathcal{X}}(x,x')} (KH)_{x'\hat{x}}
\end{aligned}$$

and therefore KH is $\epsilon d_{\mathcal{X}}$-private. □

THEOREM 2. *If a mechanism K is $d_{\mathcal{X}}$-OPTQL(π, d_Q) then it is also q-OPTPRIV(π, d_Q, d_Q) for $q = \text{QL}(K, \pi, d_Q)$.*

PROOF. Let $d_A = d_Q$. We recall from Section 2.1 that for an arbitrary mechanism M, it holds that

$$\begin{aligned}
\text{ADVERROR}(M, \pi, d_Q) &= \min_H \text{EXPDIST}(MH, \pi, d_Q) \\
&= \min_H \text{QL}(MH, \pi, d_Q)
\end{aligned}$$

which means that

$$\text{ADVERROR}(M, \pi, d_Q) \leq \text{QL}(M, \pi, d_Q) \tag{1}$$

Let K be a $d_{\mathcal{X}}$-OPTQL(π, d_Q) mechanism. Suppose that

$$\text{ADVERROR}(K, \pi, d_Q) < \text{QL}(K, \pi, d_Q)$$

This means that there is a remapping H, other than the identity, such that

$$\text{QL}(KH, \pi, d_Q) < \text{QL}(K, \pi, d_Q)$$

However, by Lemma 1 we know that KH is also $d_{\mathcal{X}}$-private, and therefore, recalling Definition 3, K would not be $d_{\mathcal{X}}$-OPTQL(π, d_Q), which is a contradiction. Therefore, we can state that

$$\text{ADVERROR}(K, \pi, d_Q) = \text{QL}(K, \pi, d_Q) \tag{2}$$

Now, in order to see that K is also q-OPTPRIV(π, d_Q, d_Q), with $q = \text{QL}(K, \pi, d_Q)$, let K' be such that

$$\text{QL}(K', \pi, d_Q) \leq \text{QL}(K, \pi, d_Q) \tag{3}$$

According to Definition 1 we need to prove that

$$\text{ADVERROR}(K', \pi, d_Q) \leq \text{ADVERROR}(K, \pi, d_Q)$$

And in fact we can see that

$$\begin{aligned}
\text{ADVERROR}(K', \pi, d_Q) &\leq \text{QL}(K', \pi, d_Q) && \text{(by (1))} \\
&\leq \text{QL}(K, \pi, d_Q) && \text{(by (3))} \\
&= \text{ADVERROR}(K, \pi, d_Q) && \text{(by (2))}
\end{aligned}$$

which concludes our proof. □

B. DUAL FORM OF THE OPTIMIZATION PROBLEM

In this section we show the dual form of the optimization problem presented in Section 3.2. We recall that the original linear program is as follows:

Minimize: $$\sum_{x,z\in\mathcal{X}} \pi_x k_{xz} d_Q(x,z)$$

Subject to:

$$k_{xz} \leq e^{\frac{\epsilon}{\delta} d_G(x,x')} k_{x'z} \qquad z\in\mathcal{X}, (x,x')\in E \qquad (1)$$

$$\sum_{x\in\mathcal{X}} k_{xz} = 1 \qquad\qquad x\in\mathcal{X} \qquad (2)$$

$$k_{xz} \geq 0 \qquad\qquad x,z\in\mathcal{X}$$

To obtain the dual form, we apply the standard technique of linear programming.

First, for the dual program we need to consider one variable for each of the constraints in the original linear program that are not constraints on single variables. Therefore we have two sets of variables:

- The variables of the form $a_{xx'z}$, with $z\in\mathcal{X}, (x,x')\in E$, corresponding to the constraints in (1).

- The variables of the form b_x, with $x\in\mathcal{X}$, corresponding to the constraints in (2).

Again applying the standard technique, we obtain the following system of constraints and objective function, that constitute the dual linear program:

Maximize: $$\sum_{x\in\mathcal{X}} b_x$$

Subject to:

$$b_x + \sum_{(x,x')\in E} (e^{\frac{\epsilon}{\delta} d_G(x,x')} a_{x'xz} - a_{xx'z}) \leq \pi_x d_Q(x,z), \quad x,z\in\mathcal{X}$$

$$a_{xx'z} \geq 0, \quad z\in\mathcal{X}, (x,x')\in E$$