# Geo-Indistinguishability: Differential Privacy for Location-Based Systems

Miguel E. Andrés [a], Nicolás E. Bordenabe [b,a], Konstantinos Chatzikokolakis [c,a], Catuscia Palamidessi [b,a]

[a] *LIX, Ècole Polytechnique*
[b] *INRIA*
[c] *CNRS*

**Abstract.** The growing popularity of location-based systems, allowing unknown/untrusted servers to easily collect huge amounts of information regarding users' location, has recently started raising serious privacy concerns. In this paper we introduce geo-indistinguishability, a formal notion of privacy for location-based systems that protects the user's exact location, while allowing approximate information – typically needed to obtain a certain desired service – to be released.

This privacy definition formalizes the intuitive notion of protecting the user's location within a radius $r$ with a level of privacy that depends on $r$, and corresponds to a generalized version of the well-known concept of *differential privacy*. Furthermore, we present a mechanism for achieving geo-indistinguishability by adding controlled random noise to the user's location.

We describe how to use our mechanism to enhance LBS applications with geo-indistinguishability guarantees without compromising the quality of the application results. Finally, we compare state-of-the-art mechanisms from the literature with ours. It turns out that, among all mechanisms independent of the prior, our mechanism offers the best privacy guarantees.

Keywords: Geolocation, Privacy Technologies, Differential privacy, Location-based services, Data sanitation, Random perturbation techniques, Planar laplacian distribution

## 1. Introduction

In recent years, the increasing availability of location information about individuals has led to a growing use of systems that record and process location data, generally referred to as "location-based systems". Such systems include (a) Location Based Services (LBSs), in which a user obtains, typically in real-time, a service related to his current location, and (b) location-data mining algorithms, used to determine points of interest and traffic patterns.

The use of LBSs, in particular, has been significantly increased by the growing popularity of mobile devices equipped with GPS chips, in combination with the increasing availability of wireless data connections. Recent studies in the US show that in 2013, 56% of the adult population of the country owns a smartphone (in comparison with 35% in 2011) [28]. Of these users. 74% use services based on their location. Examples of LBSs include mapping applications (e.g. Google Maps), Points of Interest (POI) retrieval (e.g. AroundMe), coupon/discount providers (e.g. GroupOn) and location-aware social networks (e.g. Foursquare).

While location-based systems have demonstrated to provide enormous benefits to individuals and society, the growing exposure of users' location information raises important privacy issues. First of all, location information itself may be considered as sensitive. Furthermore, it can be easily linked to a variety of other information that an individual usually wishes to protect: by collecting and processing accurate location data on a regular basis, it is possible to infer an individual's home or work location, sexual preferences, political views, religious inclinations, etc. In its extreme form, monitoring and control of an individual's location has been even described as a form of slavery [11].

Several notions of privacy for location-based systems have been proposed in the literature. In Section 2.2 we give an overview of such notions, and we discuss their shortcomings in relation to our motivating LBS application. Aiming at addressing these shortcomings, we propose a *formal privacy definition* for LBSs, as well as a randomized technique that allows a user to disclose *enough location information* to obtain the desired service, while satisfying the aforementioned privacy notion. Our proposal is based on the notion of $d$-privacy, a generalization of *differential privacy* [13] developed in [6]. $d$-privacy requires any two secrets from an arbitrary set $\mathcal{X}$ to satisfy a certain level of indistinguishability, which depends on their distance with respect to a metric $d$. In this context, we refer to $d$ as the *distinguishability metric*. It is important to note that, like differential privacy, our notion and technique abstract from the side information of the adversary, such as any prior probabilistic knowledge about the user's actual location.

As a running example, we consider a user located in Paris who wishes to query an LBS provider for nearby restaurants in a private way, i.e., by disclosing some approximate information $z$ instead of his exact location $x$. A crucial question is: what kind of privacy guarantee can the user expect in this scenario? Intuitively, the privacy level should depend on the accuracy with which an attacker can guess an individual's location from the one reported to the provider. It is logical then to aim for a distance-dependent notion of privacy, requiring points that are close in distance to each other to be *indistinguishable* from the attacker's point of view. However, we still allow the service provider to distinguish between points that are far from each other. This is exactly the kind of situation in which the notion of $d$-privacy shows to be useful. In this particular case, the privacy guarantee can also be thought as an individual having a certain level of privacy *within a radius*: we can say that the user enjoys a privacy level $l$ within a radius $r$ if any two locations at distance at most $r$ produce observations with "similar" distributions, where the "level of similarity" depends on $l$. If we take as the set of secrets $\mathcal{X}$ the set of possible locations of an individual, and if we define the distinguishability metric as $d = \frac{l}{r} d_2$ (where $d_2$ is the Euclidean distance), we can therefore give a first, intuitive definition of our location privacy notion, that we call *geo-indistinguishability*:

> A location privacy mechanism satisfies $\epsilon$-geo-indistinguishability if and only if for any radius $r > 0$, the user enjoys $\epsilon r$-privacy within $r$.

This definition implies that the user is protected within any radius $r$, but with a level $\ell = \epsilon r$ that increases with the distance. Within a short radius, for instance $r = 1$ km, $\ell$ is small, guaranteeing that the provider cannot infer the user's location within, say, the 7th arrondissement of Paris. Farther away from the user, for instance for $r = 1000$ km, $\ell$ becomes large, allowing the LBS provider to infer that with high probability the user is located in Paris instead of, say, London. Figure 1 illustrates the idea of privacy levels decreasing with the radius.

We continue by developing a mechanism to achieve geo-indistinguishability by perturbating the user's location $x$. The inspiration comes from one of the most popular approaches for differential privacy, namely the Laplacian noise. We adopt a specific planar version of the Laplace distribution, allowing to

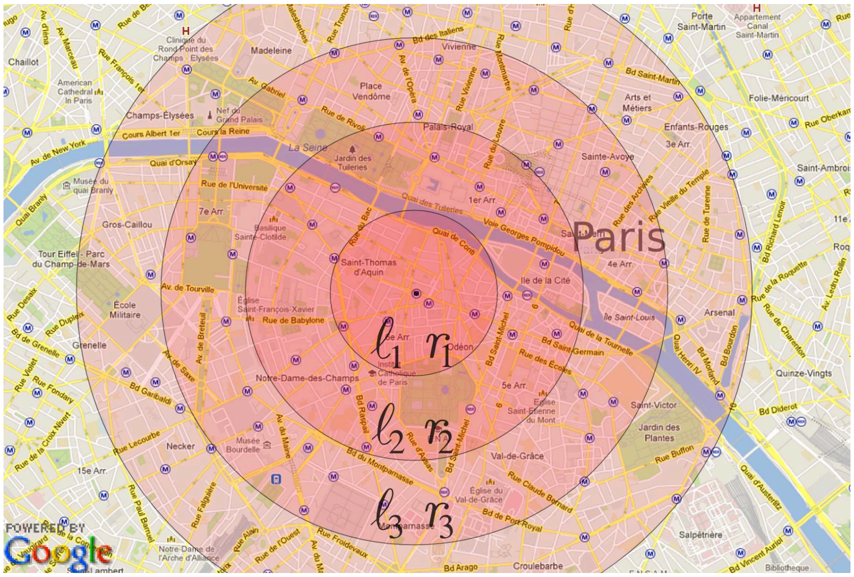



Fig. 1. Geo-indistinguishability: privacy varying with $r$.

draw points in a *geo-indistinguishable* way; moreover, we are able to do so efficiently, via a transformation to polar coordinates. However, as standard (digital) applications require a finite representation of locations, it is necessary to add a discretization step. Such operation jeopardizes the privacy guarantees, for reasons similar to the rounding effects of finite-precision operations [29]. We show how to preserve geo-indistinguishability, at the price of a degradation of the privacy level, and how to adjust the privacy parameters in order to obtain a desired level of privacy.

We then present two case studies: in the first, we describe how to use our mechanism to enhance LBS applications with geo-indistinguishability guarantees. Our proposal results in highly configurable LBS applications, both in terms of privacy and accuracy (a notion of utility/quality-of-service for LBS applications providing privacy via location perturbation techniques). Enhanced LBS applications require extra bandwidth consumption in order to provide both privacy and accuracy guarantees, thus we study how the different configurations affect the bandwidth overhead using the Google Places API [1] as reference to measure bandwidth consumption. Our experiments showed that the bandwidth overhead necessary to enhance LBS applications with very high levels of privacy and accuracy is not-prohibitive and, in most cases, negligible for modern applications. In the second case study, we show how to use our mechanism to sanitize geospatial information coming from the LODES dataset, developed by the U.S. Census Bureau, which contains (among other things) information about the home and work location of U.S. residents.

Finally, we compare our mechanism with other ones in the literature, using the privacy metric proposed in [37], which we discuss in Section 2.1. It turns our that our mechanism offers the best privacy guarantees, for the same utility, among all those which do not depend on the prior knowledge of the adversary. The advantages of the independence from the prior are obvious: first, the mechanism is designed once and for all (i.e. it does not need to be recomputed every time the adversary changes, it works also in simultaneous presence of different adversaries, etc.). Second, and even more important, it is applicable also when we do not know the prior.

*Contribution* This paper contributes to the state-of-the-art as follows:

- We show that our generalized notion of differential privacy [6], instantiated with the Euclidean metric, can be naturally applied to location privacy, and we discuss the privacy guarantees that this definition provides. (Location privacy was only briefly mentioned in [6] as a possible application.)
- We propose a mechanism to efficiently draw noise from a planar Laplace distribution, which is not trivial. Laplacians on general metric spaces were briefly discussed in [6], but no efficient method to draw from them was given. Furthermore, we cope with the crucial problems of discretization and truncation, which have been shown to pose significant threats to mechanism implementations [29].
- We describe how to use our mechanism to enhance LBS applications with geo-indistinguishability guarantees, and also to sanitize a dataset containing location information of several individuals.
- We compare our mechanism to a state-of-the-art mechanism from the literature [37] as well as a simple cloaking mechanism, obtaining favorable results.

*Road Map* In Section 2 we present the state-of-the-art on location privacy metrics, and discuss notions of location privacy from the literature, pointing out their weaknesses and strengths. In Section 3 we formalize the notion of geo-indistinguishability in three equivalent ways. We then proceed to describe a mechanism that provides geo-indistinguishability in Section 4. In Section 5 we show how to enhance LBS applications with geo-indistinguishability guarantees. Section 6 shows how to use our mechanism sanitize a dataset with location information. In Section 7 we compare the privacy guarantees of our methods with those of two other methods from the literature. Section 8 discusses related work and Section 9 concludes.

All proofs are in the appendix.

## 2. Metrics for Location Pivacy

In this section, we briefly examine various notions and mechanisms from the literature in location privacy of LBSs. We pay particular attention to the notion of *expected distance* with respect to an obfuscation mechanism, which serves as the basis for two important notions used later in this paper: the expected error of the adversary, used to quantify the location privacy provided by a mechanism, and the quality loss, used to calculate its utility.

We also explore other location privacy notions, addressing their strengths and weaknesses. This will help in defining the goals our desired privacy notion needs to satisfy.

### *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.

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 $\pi$ and a metric $d$ on $\mathcal{X}$, the expected distance between the real and the reported location is:

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

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

$$\mathrm{QL}(K, \pi, d_Q) = \textsc{ExpDist}(K, \pi, d_Q)$$

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 is to consider a Bayesian adversary with some prior information $\pi$, trying to remap $z$ back to a guessed location $\hat{x}$. A remapping strategy can be modelled by a probabilistic function $H$, 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 [36,37,22]:

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

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 $\pi$, 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. $\mathrm{QL}(K, \pi, d_Q) \leq q$, *and*
2. *for all mechanisms $K'$,* $\mathrm{QL}(K', \pi, d_Q) \leq q$ *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 [37], providing a method to construct such a mechanism for any $q, \pi, d_A, d_Q$, by solving a zero-sum Bayesian Stackelberg game with a properly constructed linear program.

It is worth noting that this privacy notion and the obfuscation mechanisms based on it are explicitly defined in terms of the adversary's side information. This implies that location-obfuscation mechanisms based on this notion assume that the attacker have some particular kind of side-information (for instance, past location traces of the user), and therefore the definition is only satisfied for this limited class of adversaries.

### *2.2. Other ways to measure location privacy*

#### *$k$-anonymity*

The notion of $k$-anonymity is the most widely used definition of privacy for location-based systems in the literature. Many systems in this category [20,18,30] aim at protecting the user's *identity*, requiring that the attacker cannot infer which user is executing the query, among a set of $k$ different users. Such systems are outside the scope of our problem, since we are interested in protecting the user's *location*.

On the other hand, $k$-anonymity has also been used to protect the user's location (sometimes called $l$-diversity in this context), requiring that it is indistinguishable among a set of $k$ points (often required to share some semantic property). One way to achieve this is through the use of *dummy locations* [24,34]. This technique involves generating $k-1$ properly selected dummy points, and performing $k$ queries to the service provider, using the real and dummy locations. Another method for achieving $k$-anonymity is through *cloaking* [4,12,40]. This involves creating a cloaking region that includes $k$ points sharing some property of interest, and then querying the service provider for this cloaking region.

Even when side knowledge does not explicitly appear in the definition of $k$-anonymity, a system cannot be proven to satisfy this notion unless assumptions are made about the attacker's side information. For example, dummy locations are only useful if they look equally likely to be the real location from the point of view of the attacker. Any side information that allows to rule out any of those points, as having low probability of being the real location, would immediately violate the definition.

Counter-measures are often employed to avoid this issue: for instance, [24] takes into account concepts such as ubiquity, congestion and uniformity for generating dummy points, in an effort to make them look realistic. Similarly, [40] takes into account the user's side information to construct a cloaking region. Such counter-measures have their own drawbacks: first, they complicate the employed techniques, also requiring additional data to be taken into account (for instance, precise information about the environment or the location of nearby users), making their application in real-time by a handheld device challenging. Moreover, the attacker's actual side information might simply be inconsistent with the assumptions being made. A detailed study of the flaws of $k$-anonymity as a framework for location privacy have also been studied in [38].

As a result, notions that abstract from the attacker's side information, such as differential privacy, have been growing in popularity in recent years, compared to $k$-anonymity-based approaches.

*Differential Privacy*

Differential privacy has also been used in the context of location privacy. In the work of [27], it is shown that a synthetic data generation technique can be used to publish statistical information about commuting patterns in a differentially private way. In [21], a quadtree spatial decomposition technique is used to ensure differential privacy in a database with location pattern mining capabilities.

As shown in the aforementioned works, differential privacy can be successfully applied in cases where *aggregate* information about several users is published. On the other hand, the nature of this notion makes it poorly suitable for applications in which only a single individual is involved, such as our motivating scenario. The secret in this case is the location of a single user. Thus, differential privacy would require that any change in that location should have negligible effect on the published output, making it impossible to communicate any useful information to the service provider.

To overcome this issue, Dewri [10] proposes a mix of differential privacy and $k$-anonymity, by fixing an anonymity set of $k$ locations and requiring that the probability to report the same obfuscated location $z$ from any of these $k$ locations should be similar (up to $e^{\epsilon}$). This property is achieved by adding Laplace noise to each Cartesian coordinate independently. There are however two problems with this definition: first, the choice of the anonymity set crucially affects the resulting privacy; outside this set no privacy is guaranteed at all. Second, the property itself is rather weak; reporting the geometric median (or any deterministic function) of the $k$ locations would satisfy the same definition, although the privacy guarantee would be substantially lower than using Laplace noise.

Nevertheless, Dewri's intuition of using Laplace noise[1] for location privacy is valid, and [10] provides extensive experimental analysis supporting this claim.

[1] The planar Laplace distribution that we use later in this paper, however, is different from the distribution obtained by adding Laplace noise to each Cartesian coordinate, and has better differential privacy properties (c.f. Section 4).

*Approach-specific location-privacy metrics*

There are also other location-privacy definitions that can be found in the literature, usually specific to some particular obfuscation mechanism. [9] proposes a location cloaking mechanism, and focuses on the evaluation of Location-based Range Queries. The degree of privacy is measured by the size of the cloak (also called *uncertainty region*), and by the coverage of sensitive regions, which is the ratio between the area of the cloak and the area of the regions inside the cloak that the user considers to be sensitive. In order to deal with the side-information that the attacker may have, ad-hoc solutions are proposed, like patching cloaks to enlarge the uncertainty region or delaying requests. Both solutions may cause a degradation in the quality of service.

In [3], the real location of the user is assumed to have some level of inaccuracy, due to the specific sensing technology or to the environmental conditions. Different obfuscation techniques are then used to increase this inaccuracy in order to achieve a certain level of privacy. This level of privacy is computed as (the opposite of) the *relevance* of the location measurement. Relevance is defined as the ratio between the accuracy before and after the application of the obfuscation techniques.

Similar to the case of $k$-anonymity, both privacy metrics mentioned above make implicit assumptions about the adversary's side information. This may imply a violation of the privacy definition in a scenario where the adversary has some knowledge (maybe probabilistic) about the user's real location.

*Transformation-based approaches*

A number of approaches for location privacy are radically different from the ones mentioned so far. Instead of cloaking the user's location, they aim at making it completely invisible to the service provider. This is achieved by transforming all data to a different space, usually employing cryptographic techniques, so that they can be mapped back to spatial information only by the user [23,19]. The data stored in the provider, as well as the location send by the user are encrypted. Then, using techniques from *private information retrieval*, the provider can return information about the encrypted location, without ever discovering which actual location it corresponds to.

A drawback of these techniques is that they are computationally demanding, making it difficult to implement them in a handheld device. Moreover, they require the provider's data to be encrypted, making it impossible to use existing providers, such as Google Maps, which have access to the real data.

## 3. Geo-Indistinguishability

In this section we formalize our notion of geo-indistinguishability, as an instance of $d$-privacy. As already discussed in the introduction, the main idea behind this notion is that, for any radius $r > 0$, the user enjoys $\epsilon r$-privacy within $r$, i.e. the level of privacy is proportional to the radius. Note that the parameter $\epsilon$ corresponds to the level of privacy at one unit of distance. For the user, a simple way to specify his privacy requirements is by a tuple $(\ell, r)$, where $r$ is the radius he is mostly concerned with and $\ell$ is the privacy level he wishes *for that radius*. In this case, it is sufficient to require $\epsilon$-geo-indistinguishability for $\epsilon = \ell/r$; this will ensure a level of privacy $\ell$ within $r$, and a proportionally selected level for all other radii.

So far we kept the discussion on an informal level by avoiding to explicitly define what $\ell$-privacy within $r$ means. In the remaining of this section we give a formal definition, as well as two characterizations which clarify the privacy guarantees provided by geo-indistinguishability.

*Probabilistic model* We first introduce a simple model used in the rest of the paper. We start with a set $\mathcal{X}$ of *points of interest*, typically the user's possible locations. Moreover, let $\mathcal{Z}$ be a set of possible *reported values*, which in general can be arbitrary, allowing to report obfuscated locations, cloaking regions, sets of locations, etc. However, to simplify the discussion, we sometimes consider $\mathcal{Z}$ to also contain spatial points, assuming an operational scenario of a user located at $x \in \mathcal{X}$ and communicating to the attacker a randomly selected location $z \in \mathcal{Z}$ (e.g. an obfuscated point).

Probabilities come into place in two ways. First, the attacker might have side information about the user's location, knowing, for example, that he is likely to be visiting the Eiffel Tower, while unlikely to be swimming in the Seine river. The attacker's side information can be modeled by a *prior* distribution $\pi$ on $\mathcal{X}$, where $\pi(x)$ is the probability assigned to the location $x$.

Second, the selection of a reported value in $\mathcal{Z}$ is itself probabilistic; for instance, $z$ can be obtained by adding random noise to the actual location $x$ (a technique used in Section 4). As mentioned in Section 2.1, a mechanism $K$ is a probabilistic function for selecting a reported value; i.e. $K$ is a function assigning to each location $x \in \mathcal{X}$ a probability distribution on $\mathcal{Z}$, where $K(x)(Z)$ is the probability that the reported point belongs to the set $Z \subseteq \mathcal{Z}$, when the user's location is $x$.[2] Starting from $\pi$ and using Bayes' rule, each observation $Z \subseteq \mathcal{Z}$ of a mechanism $K$ induces a *posterior* distribution $\sigma = \mathbf{Bayes}(\pi, K, Z)$ on $\mathcal{X}$, defined as

$$\sigma(x) = \frac{K(x)(Z)\pi(x)}{\sum_{x'} K(x')(Z)\pi(x')}$$

We define the *multiplicative distance* between two distributions $\sigma_1, \sigma_2$ on some set $\mathcal{S}$ as

$$d_{\mathcal{P}}(\sigma_1, \sigma_2) = \sup_{S \subseteq \mathcal{S}} |\ln \tfrac{\sigma_1(S)}{\sigma_2(S)}|$$

with the convention that $|\ln \frac{\sigma_1(S)}{\sigma_2(S)}| = 0$ if both $\sigma_1(S), \sigma_2(S)$ are zero and $\infty$ if only one of them is zero.

### *3.1. Definition*

We are now ready to state our definition of geo-indistinguishability. Intuitively, a privacy requirement is a constraint on the distributions $K(x), K(x')$ produced by two different points $x, x'$. Let $d_2(\cdot, \cdot)$ denote the Euclidean metric. Enjoying $\ell$-privacy within $r$ means that for any $x, x'$ s.t. $d_2(x, x') \leq r$, the distance $d_{\mathcal{P}}(K(x), K(x'))$ between the corresponding distributions should be at most $l$. Then, requiring $\epsilon r$-privacy for all radii $r$, forces the two distributions to be similar for locations close to each other, while relaxing the constraint for those far away from each other, allowing a service provider to distinguish points in Paris from those in London.

**Definition 2** (geo-indistinguishability)**.** *A mechanism $K$ satisfies $\epsilon$-geo-indistinguishability iff for all $x, x' \in \mathcal{X}$:*

$$d_{\mathcal{P}}(K(x), K(x')) \leq \epsilon d_2(x, x')$$

Equivalently, the definition can be formulated as $K(x)(Z) \leq e^{\epsilon d_2(x,x')} K(x')(Z)$ for all $x, x' \in \mathcal{X}, Z \subseteq \mathcal{Z}$. Note that for all points $x'$ within a radius $r$ from $x$, the definition forces the corresponding distributions to be at most $\epsilon r$ distant.

[2]For simplicity we assume distributions on $\mathcal{X}$ to be discrete, but allow those on $\mathcal{Z}$ to be continuous (c.f. Section 4). All sets to which probability is assigned are implicitly assumed to be measurable.

The above definition is very similar to the one of differential privacy, which requires $d_{\mathcal{P}}(K(x), K(x')) \leq \epsilon d_h(x, x')$, where $d_h$ is the Hamming distance between databases $x, x'$, i.e. the number of individuals in which they differ. In fact, geo-indistinguishability is an instance of a generalized variant of differential privacy, using an arbitrary metric between secrets. This generalized formulation has been known for some time: for instance, [31] uses it to perform a compositional analysis of standard differential privacy for functional programs, while [15] uses metrics between individuals to define "fairness" in classification. On the other hand, the usefulness of using different metrics to achieve different privacy goals and the semantics of the privacy definition obtained by different metrics have only recently started to be studied [6]. This paper focuses on location-based systems and is, to our knowledge, the first work considering privacy under the Euclidean metric, which is a natural choice for spatial data.

Note that in our scenario, using the Hamming metric of standard differential privacy – which aims at completely protecting the value of an individual – would be too strong, since the only information is the location of a single individual. Nevertheless, we are not interested in completely hiding the user's location, since some approximate information needs to be revealed in order to obtain the required service. Hence, using a privacy level that depends on the Euclidean distance between locations is a natural choice.

*A note on the unit of measurement* It is worth noting that, since $\epsilon$ corresponds to the privacy level for one unit of distance, it is affected by the unit in which distances are measured. For instance, assume that $\epsilon = 0.1$ and distances are measured in meters. The level of privacy for points one kilometer away is $1000\epsilon$, hence changing the unit to kilometers requires to set $\epsilon = 100$ in order for the definition to remain unaffected. In other words, if $r$ is a physical quantity expressed in some unit of measurement, then $\epsilon$ has to be expressed in the inverse unit.

### *3.2. Characterizations*

In this section we state two characterizations of geo-indistinguishability, obtained from the corresponding results of [6] (for general metrics), which provide intuitive interpretations of the privacy guarantees offered by geo-indistinguishability.

*Adversary's conclusions under hiding*

The first characterization uses the concept of a *hiding function* $\phi : \mathcal{X} \to \mathcal{X}$. The idea is that $\phi$ can be applied to the user's actual location before the mechanism $K$, so that the latter has only access to a hidden version $\phi(x)$, instead of the real location $x$. A mechanism $K$ with hiding applied is simply the composition $K \circ \phi$. Intuitively, a location remains private if, regardless of his side knowledge (captured by his prior distribution), an adversary draws the same conclusions (captured by his posterior distribution), regardless of whether hiding has been applied or not. However, if $\phi$ replaces locations in Paris with those in London, then clearly the adversary's conclusions will be greatly affected. Hence, we require that the effect on the conclusions depends on the maximum distance $d(\phi) = \sup_{x \in \mathcal{X}} d(x, \phi(x))$ between the real and hidden location.

**Theorem 1.** *A mechanism $K$ satisfies $\epsilon$-geo-indistinguishability iff for all $\phi : \mathcal{X} \to \mathcal{X}$, all priors $\pi$ on $\mathcal{X}$, and all $Z \subseteq \mathcal{Z}$:*

$$d_{\mathcal{P}}(\sigma_1, \sigma_2) \leq 2\epsilon d(\phi) \qquad \textit{where} \qquad \begin{aligned} \sigma_1 &= \mathbf{Bayes}(\pi, K, Z) \\ \sigma_2 &= \mathbf{Bayes}(\pi, K \circ \phi, Z) \end{aligned}$$

Note that this is a natural adaptation of a well-known interpretation of standard differential privacy, stating that the attacker's conclusions are similar, regardless of his side knowledge, and regardless of whether an individual's real value has been used in the query or not. This corresponds to a hiding function $\phi$ removing the value of an individual.

Note also that the above characterization compares two *posterior* distributions. Both $\sigma_1, \sigma_2$ can be substantially different than the initial knowledge $\pi$, which means that an adversary does learn some information about the user's location.

*Knowledge of an informed attacker*

A different approach is to measure how much the adversary learns about the user's location, by comparing his prior and posterior distributions. However, since some information is allowed to be revealed by design, these distributions can be far apart. Still, we can consider an *informed* adversary who already knows that the user is located within a set $N \subseteq \mathcal{X}$. Let $d(N) = \sup_{x,x' \in N} d(x, x')$ be the maximum distance between points in $x$. Intuitively, the user's location remains private if, regardless of his prior knowledge within $N$, the knowledge obtained by such an informed adversary should be limited by a factor depending on $d(N)$. This means that if $d(N)$ is small, i.e. the adversary already knows the location with some accuracy, then the information that he obtains is also small, meaning that he cannot improve his accuracy. Denoting by $\pi_{|N}$ the distribution obtained from $\pi$ by restricting to $N$ (i.e. $\pi_{|N}(x) = \pi(x|N)$), we obtain the following characterization:

**Theorem 2.** *A mechanism $K$ satisfies $\epsilon$-geo-indistinguishability iff for all $N \subseteq \mathcal{X}$, all priors $\pi$ on $\mathcal{X}$, and all $Z \subseteq \mathcal{Z}$:*

$$d_{\mathcal{P}}(\pi_{|N}, \sigma_{|N}) \leq \epsilon d(N) \qquad \textit{where} \qquad \sigma = \mathbf{Bayes}(\pi, K, Z)$$

Note that this is a natural adaptation of a well-known interpretation of standard differential privacy, stating that in informed adversary who already knows all values except individual's $i$, gains no extra knowledge from the reported answer, regardless of side knowledge about $i$'s value [16].

*Abstracting from side information*

A major difference of geo-indistinguishability, compared to similar approaches from the literature, is that it abstracts from the side information available to the adversary, i.e. from the prior distribution. This is a subtle issue, and often a source of confusion, thus we would like to clarify what "abstracting from the prior" means. The goal of a privacy definition is to restrict the information *leakage* caused by the observation. Note that the lack of leakage does not mean that the user's location cannot be inferred (it could be inferred by the prior alone), but instead that the adversary's knowledge does not increase *due to the observation*.

However, in the context of LBSs, no privacy definition can ensure a small leakage under any prior, and at the same time allow reasonable utility. Consider, for instance, an attacker who knows that the user is located at some airport, but not which one. The attacker's prior knowledge is very limited, still any useful LBS query should reveal at least the user's city, from which the exact location (i.e. the city's airport) can be inferred. Clearly, due to the side information, the leakage caused by the observation is high.

So, since we cannot eliminate leakage under any prior, how can we give a reasonable privacy definition without restricting to a particular one? First, we give a formulation (Definition 2) which does not involve the prior at all, allowing to verify it without knowing the prior. At the same time, we give two characterizations which explicitly quantify over all priors, shedding light on how the prior affects the privacy guarantees.

Finally, we should point out that differential privacy abstracts from the prior in exactly the same way. Contrary to what is sometimes believed, the user's value is *not protected* under any prior information. Recalling the well-known example from [13], if the adversary knows that Terry Gross is two inches shorter than the average Lithuanian woman, then he can accurately infer the height, even if the average is release in a differentially private way (in fact no useful mechanism can prevent this leakage). Differential privacy does ensure that her risk is the same whether she participates in the database or not, but this might me misleading: it does not imply the lack of leakage, only that it will happen anyway, whether she participates or not!

### *3.3. Protecting location sets*

So far, we have assumed that the user has a single location that he wishes to communicate to a service provider in a private way (typically his current location). In practice, however, it is common for a user to have multiple points of interest, for instance a set of past locations or a set of locations he frequently visits. In this case, the user might wish to communicate to the provider some information that depends on all points; this could be either the whole set of points itself, or some aggregate information, for instance their centroid. As in the case of a single location, privacy is still a requirement; the provider is allowed to obtain only approximate information about the locations, their exact value should be kept private. In this section, we discuss how $\epsilon$-geo-indistinguishability extends to the case where the secret is a tuple of points $\mathbf{x} = (x_1, \ldots, x_n)$.

Similarly to the case of a single point, the notion of distance is crucial for our definition. We define the distance between two tuples of points $\mathbf{x} = (x_1, \ldots, x_n), \mathbf{x}' = (x'_1, \ldots, x'_n)$ as:

$$d_\infty(\mathbf{x}, \mathbf{x}') = \max_i d(x_i, x'_i)$$

Intuitively, the choice of metric follows the idea of reasoning within a radius $r$: when $d_\infty(\mathbf{x}, \mathbf{x}') \leq r$, it means that all $x_i, x'_i$ are within distance $r$ from each other. All definitions and results of this section can be then directly applied to the case of multiple points, by using $d_\infty$ as the underlying metric. Enjoying $\ell$-privacy within a radius $r$ means that two tuples at most $r$ away from each other, should produce distributions at most $\epsilon r$ apart.

*Reporting the whole set*

A natural question then to ask is how we can obfuscate a tuple of points, by independently applying an existing mechanism $K_0$ to each individual point, and report the obfuscated tuple. Starting from a tuple $\mathbf{x} = (x_1, \ldots, x_n)$, we independently apply $K_0$ to each $x_i$ obtaining a reported point $z_i$, and then report the tuple $\mathbf{z} = (z_1, \ldots, z_n)$. Thus, the probability that the combined mechanism $K$ reports $\mathbf{z}$, starting from $\mathbf{x}$, is the product of the probabilities to obtain each point $z_i$, starting from the corresponding point $x_i$, i.e. $K(\mathbf{x})(\mathbf{z}) = \prod_i K_0(x_i)(z_i)$.

The next question is what level of privacy does $K$ satisfy. For simplicity, consider a tuple of only two points $(x_1, x_2)$, and assume that $K_0$ satisfies $\epsilon$-geo-indistinguishability. At first look, one might expect the combined mechanism $K$ to also satisfy $\epsilon$-geo-indistinguishability, however this is not the case. The problem is that the two points might be *correlated*, thus an observation about $x_1$ will reveal information about $x_2$ and vice versa. Consider, for instance, the extreme case in which $x_1 = x_2$. Having two observations about the same point reduces the level of privacy, thus we cannot expect the combined mechanism to provide the same level of privacy.

Still, if $K_0$ satisfies $\epsilon$-geo-indistinguishability, then $K$ can be shown to satisfy $n\epsilon$-geo-indistinguishability, i.e. a level of privacy that scales linearly with $n$. Due to this scalability issue, the technique of independently applying a mechanism to each point is only useful when the number of points is small. Still, this is sufficient for some applications, such as the case study of Section 5. Note, however, that this technique is by no means the best we can hope for: similarly to standard differential privacy [5,32], better results could be achieved by adding noise to the whole tuple $\mathbf{x}$, instead of each individual point.

*Dealing with location traces*

As discussed above, if we need to add noise to a tuple of $n$ points, then doing it by applying an $\epsilon$-geo-indistinguishable mechanism independently to each point in the tuple guarantees $n\epsilon$-geo-indistinguishability. However, it is clear that in this case the level of privacy decreases fast with respect to the number of points in the tuple.

A location trace is a particular case of tuple, in which not only subsequent points are highly correlated, but also the noise to each points must be added dynamically, i.e. we cannot just add noise to the tuple as a whole. Besides, location traces usually contain a large number of points, making the previously mentioned approach of adding independent noise ineffective.

To deal with this particular scenario, the authors of [7] have proposed a method to add noise to location traces by means of a prediction function. The basic idea is that, whenever a new point is added to the trace and has to be obfuscated, the mechanism first generates a fake point by "predicting" the direction in which the user has moved (i.e. in this step, no randomization is used). This point is then tested to determine if it falls "too far" from the real location. Then, only if this is the case, a new point is generated by means of a geo-indistinguishable mechanism. This way, the authors succeed in obfuscating traces with a high number of locations with a good level of privacy.

*Reporting an aggregate location*

Another interesting case is when we need to report some aggregate information obtained by $\mathbf{x}$, for instance the centroid of the tuple. In general we might need to report the result of a query $f : \mathcal{X}^n \to \mathcal{X}$. Similarly to the case of standard differential privacy, we can compute the real answer $f(\mathbf{x})$ and the add noise by applying a mechanism $K$ to it. If $f$ is $\Delta$-sensitive wrt $d, d_\infty$, meaning that $d(f(\mathbf{x}), f(\mathbf{x}')) \leq \Delta d_\infty(\mathbf{x}, \mathbf{x}')$ for all $\mathbf{x}, \mathbf{x}'$, and $K$ satisfies geo-indistinguishability, then the composed mechanism $K \circ f$ can be shown to satisfy $\Delta\epsilon$-geo-indistinguishability.

Note that when dealing with aggregate data, standard differential privacy becomes a viable option. However, one needs to also examine the loss of utility caused by the added noise. This highly depends on the application: differential privacy is suitable for publishing aggregate queries with *low sensitivity*, meaning that changes in a single individual have a relatively small effect on the outcome. On the other hand, location information often has high sensitivity. A trivial example is the case where we want to publish the complete tuple of points. But sensitivity can be high even for aggregate information: consider the case of publishing the centroid of 5 users located anywhere in the world. Modifying a single user can hugely affect their centroid, thus achieving differential privacy would require so much noise that the result would be useless. For geo-indistinguishability, on the other hand, one needs to consider the distance between points when computing the sensitivity. In the case of the centroid, a small (in terms of distance) change in the tuple has a small effect on the result, thus geo-indistinguishability can be achieved with much less noise.

## 4. The Planar Laplace Mechanism

In this section we present a method to generate noise so to satisfy geo-indistinguishability, based on an instance of the corresponding Laplace mechanism presented in [6]. We model the location domain as a discrete[3] Cartesian plane with the standard notion of Euclidean distance. This model can be considered a good approximation of the Earth surface when the area of interest is not too large. In the rest of the section we develop our mechanism according to the following plan:

(a) First, we define a mechanism to achieve geo-indistinguishability in the ideal case of the continuous plane. For each actual location, this mechanism should generate a random point in a way that satisfies geo-indistinguishability on $\mathbb{R}^2$.
(b) Then, we discretize the mechanism by remapping each point generated according to (a) to the closest point in the discrete domain.
(c) Finally, we truncate the mechanism, so to report only points within the limits of the considered area.

### 4.1. A mechanism for the continuous plane

Following the above plan, we start by defining a mechanism for geo-indistinguishability on the continuous plane. The idea is that whenever the actual location is $x \in \mathbb{R}^2$, we report, instead, a point $z \in \mathbb{R}^2$ generated randomly according to the noise function. The latter needs to be such that the probabilities of reporting a point in a certain (infinitesimal) area around $z$, when the actual locations are $x$ and $x'$ respectively, differs at most by a multiplicative factor $e^{-\epsilon\, d_2(x,x')}$. We can achieve this property by requiring that the probability of generating a point in the area around $z$ decreases exponentially with the distance from the actual location $x$. In a linear space this is exactly the behavior of the Laplace distribution, whose probability density function (pdf) is $\epsilon/2\, e^{-\epsilon\,|z-\mu|}$. This distribution has been used in the literature to add noise to query results on statistical databases, with $\mu$ set to be the actual answer, and it can be shown to satisfy $\epsilon$-differential privacy [14].

There are two possible definitions of Laplace distribution on higher dimensions (multivariate Laplacians). The first one, investigated in [26], and used also in [16], is obtained from the standard Laplacian by replacing $|z-\mu|$ with $d(z,\mu)$. The second way consists in generating each Cartesian coordinate independently, according to a linear Laplacian. For reasons that will become clear in the next paragraph, we adopt the first approach.

*The probability density function*

Given the parameter $\epsilon \in \mathbb{R}^+$, and the actual location $x \in \mathbb{R}^2$, the pdf of our noise mechanism, on any other point $z \in \mathbb{R}^2$, is:

$$D_\epsilon(x)(z) = \frac{\epsilon^2}{2\,\pi}\, e^{-\epsilon\, d_2(x,z)} \tag{1}$$

where $\epsilon^2/2\,\pi$ is a normalization factor. We call this function *planar Laplacian centered at $x$*. The corresponding distribution is illustrated in Figure 2. It is possible to show that (i) the projection of a planar Laplacian on any vertical plane passing by the center gives a (scaled) linear Laplacian, and (ii) the corresponding mechanism satisfies $\epsilon$-geo-indistinguishability. These two properties would not be satisfied by the second approach to the multivariate Laplacian.

[3]For applications with digital interface the domain of interest is discrete, since the representation of the coordinates of the points is necessarily finite.

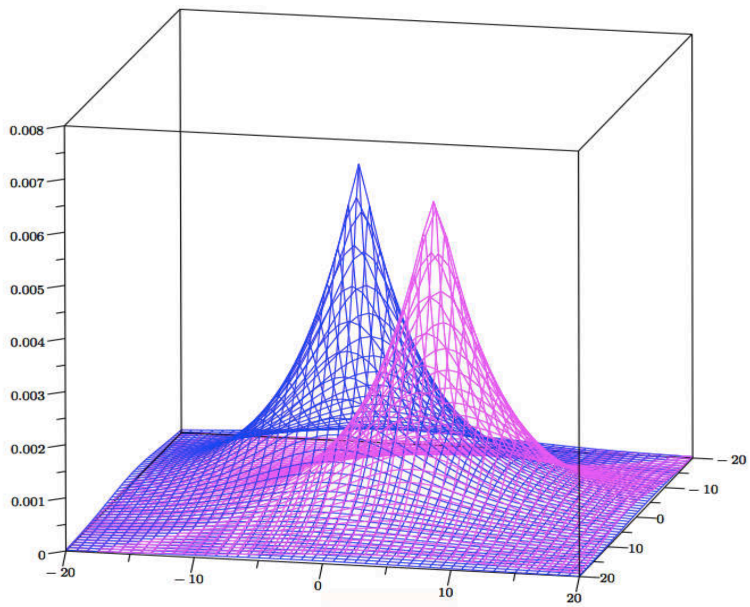



Fig. 2. The pdf of two planar Laplacians, centered at $(-2,-4)$ and at $(5,3)$ respectively, with $\epsilon = 1/5$.

*Drawing a random point*

We illustrate now how to draw a random point from the pdf defined in (1). First of all, we note that the pdf of the planar Laplacian depends only on the distance from $x$. It will be convenient, therefore, to switch to a system of polar coordinates with origin in $x$. A point $z$ will be represented as a point $(r,\theta)$, where $r$ is the distance of $z$ from $x$, and $\theta$ is the angle that the line $z\,x$ forms with respect to the horizontal axis of the Cartesian system. Following the standard transformation formula, the pdf of the *polar Laplacian* centered at the origin $(x)$ is:

$$D_\epsilon(r,\theta) = \frac{\epsilon^2}{2\,\pi}\, r\, e^{-\epsilon\, r} \tag{2}$$

We note now that the polar Laplacian defined above enjoys a property that is very convenient for drawing in an efficient way: *the two random variables that represent the radius and the angle are independent*. Namely, the pdf can be expressed as the product of the two marginals. In fact, let us denote these two random variables by $R$ (the radius) and $\Theta$ (the angle). The two marginals are:

$$D_{\epsilon,R}(r) = \int_0^{2\pi} D_\epsilon(r,\theta)\, d\theta \;=\; \epsilon^2\, r\, e^{-\epsilon\, r}$$

$$D_{\epsilon,\Theta}(\theta) = \int_0^{\infty} D_\epsilon(r,\theta)\, dr \;=\; \frac{1}{2\,\pi}$$

Hence we have $D_\epsilon(r,\theta) = D_{\epsilon,R}(r)\ D_{\epsilon,\Theta}(\theta)$. Note that $D_{\epsilon,R}(r)$ corresponds to the pdf of the *gamma distribution* with shape 2 and scale $1/\epsilon$.

Thanks to the fact that $R$ and $\Theta$ are independent, in order to draw a point $(r,\theta)$ from $D_\epsilon(r,\theta)$ it is sufficient to draw separately $r$ and $\theta$ from $D_{\epsilon,R}(r)$ and $D_{\epsilon,\Theta}(\theta)$ respectively.

Since $D_{\epsilon,\Theta}(\theta)$ is constant, drawing $\theta$ is easy: it is sufficient to generate $\theta$ as a random number in the interval $[0,2\pi)$ with uniform distribution.

**Drawing a point $(r, \theta)$ from the polar Laplacian**

1. draw $\theta$ uniformly in $[0, 2\pi)$
2. draw $p$ uniformly in $[0, 1)$ and set $r = C_\epsilon^{-1}(p)$

Fig. 3. Method to generate Laplacian noise.

We now show how to draw $r$. Following standard lines, we consider the cumulative distribution function (cdf) $C_\epsilon(r)$:

$$C_\epsilon(r) \;=\; \int_0^r D_{\epsilon,R}(\rho)d\rho \;=\; 1 - (1 + \epsilon\, r)\, e^{-\epsilon\, r}$$

Intuitively, $C_\epsilon(r)$ represents the probability that the radius of the random point falls between 0 and $r$. Finally, we generate a random number $p$ with uniform probability in the interval $[0, 1)$, and we set $r = C_\epsilon^{-1}(p)$. Note that

$$C_\epsilon^{-1}(p) = -\tfrac{1}{\epsilon}\big(W_{-1}(\tfrac{p-1}{e}) + 1\big)$$

where $W_{-1}$ is the Lambert W function (the $-1$ branch), which can be computed efficiently and is implemented in several numerical libraries (MATLAB, Maple, GSL, ...).

### *4.2. Discretization*

We discuss now how to approximate the Laplace mechanism on a grid $\mathcal{G}$ of discrete Cartesian coordinates. Let us recall the points (a) and (b) of the plan, in light of the development so far: given the actual location $x_0$, report the point $x$ in $\mathcal{G}$ obtained as follows:

(a) first, draw a point $(r, \theta)$ following the method in Figure 3,
(b) then, remap $(r, \theta)$ to the closest point $x$ on $\mathcal{G}$.

We will denote by $K_\epsilon : \mathcal{G} \to \mathcal{P}(\mathcal{G})$ the above mechanism. In summary, $K_\epsilon(x)(z)$ represents the probability of reporting the point $z$ when the actual point is $x$.

It is not obvious that the discretization preserves geo-indistinguishability, due to the following problem: in principle, each point $x$ in $\mathcal{G}$ should gather the probability of the set of points for which $x$ is the closest point in $\mathcal{G}$, namely

$$R(x) = \{y \in \mathbb{R}^2 \mid \forall x' \in \mathcal{G}.\; d(x, y) \leq d(x', y)\}$$

However, due to the finite precision of the machine, the noise generated according to (a) is already discretized in accordance with the polar system. Let $\mathcal{W}$ denote the discrete set of points actually generated in (a). Each of those points $(r, \theta)$ is drawn with the probability of the area between $r$, $r + \delta_r$, $\theta$ and $\theta + \delta_\theta$, where $\delta_r$ and $\delta_\theta$ denote the precision of the machine in representing the radius and the angle respectively. Hence, step (b) generates a point $x$ in $\mathcal{G}$ with the probability of the set $R_\mathcal{W}(x) = R(x) \cap \mathcal{W}$. This introduces some irregularity in the mechanism, because the region associated to $R_\mathcal{W}(x)$ has a different shape and area depending on the position of $x$ relatively to $x_0$. The situation is illustrated in Figure 4 with $R_0 = R_\mathcal{W}(x_0)$ and $R_1 = R_\mathcal{W}(x_1)$.

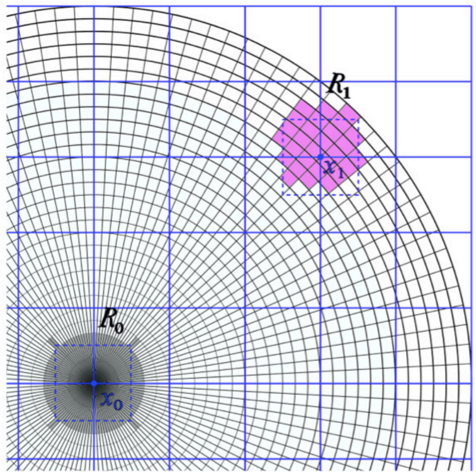



Fig. 4. Remapping the points in polar coordinates to points in the grid.

*Geo-indistinguishability of the discretized mechanism*

We now analyze the privacy guarantees provided by our discretized mechanism. We show that the discretization preserves geo-indistinguishability, at the price of a degradation of the privacy parameter $\epsilon$.

For the sake of generality we do not require the step units along the two dimensions of $\mathcal{G}$ to be equal. We will call them *grid units*, and will denote by $u$ and $v$ the smaller and the larger unit, respectively. We recall that $\delta_\theta$ and $\delta_r$ denote the precision of the machine in representing $\theta$ and $r$, respectively. We assume that $\delta_r \leq r_{\max}\delta_\theta$. The following theorem states the geo-indistinguishability guarantees provided by our mechanism: $K_{\epsilon'}$ satisfies $\epsilon$-geo-indistinguishability, within a range $r_{\max}$, provided that $\epsilon'$ is chosen in a suitable way that depends on $\epsilon$, on the length of the step units of $\mathcal{G}$, and on the precision of the machine.

**Theorem 3.** *Assume $r_{\max} < {}^{u}/_{\delta_\theta}$, and let $q = {}^{u}/_{r_{\max}\delta_\theta}$. Let $\epsilon, \epsilon' \in \mathbb{R}^+$ such that*

$$\epsilon' + \frac{1}{u} \ln \frac{q + 2\, e^{\epsilon' u}}{q - 2\, e^{\epsilon' u}} \leq \epsilon$$

*Then $K_{\epsilon'}$ provides $\epsilon$-geo-indistinguishability within the range of $r_{\max}$. Namely, if $d(x, z), d(x', z) \leq r_{\max}$ then:*

$$K_{\epsilon'}(x)(z) \leq e^{\epsilon\, d(x,x')} K_{\epsilon'}(x')(z).$$

The difference between $\epsilon'$ and $\epsilon$ represents the additional noise needed to compensate the effect of discretization. Note that $r_{\max}$, which determines the area in which $\epsilon$-geo-indistinguishability is guaranteed, must be chosen in such a way that $q > 2\, e^{\epsilon' u}$. Furthermore there is a trade-off between $\epsilon'$ and $r_{\max}$: If we want $\epsilon'$ to be close to $\epsilon$ then we need $q$ to be large. Depending on the precision, this may or may not imply a serious limit on $r_{\max}$. Vice versa, if we want $r_{\max}$ to be large then, depending on the precision, $\epsilon'$ may need to be significantly smaller than $\epsilon$, and furthermore we may have a constraint on the minimum possible value for $\epsilon$, which means that we may not have the possibility of achieving an arbitrary level of geo-indistinguishability.

Figure 5 shows how the additional noise varies depending on the precision of the machine. In this figure, $r_{\max}$ is set to be $10^2$ km, and we consider the cases of double precision (16 significant digits, i.e.,

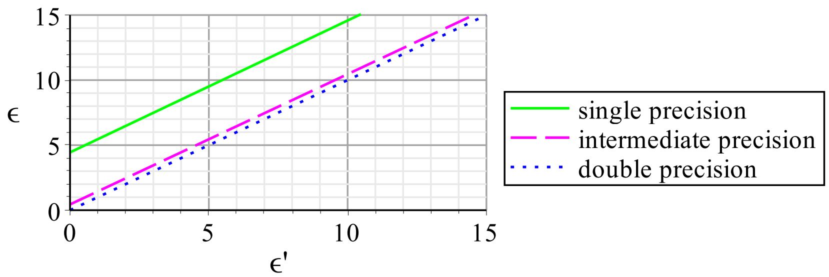



Fig. 5. The relation between $\epsilon$ and $\epsilon'$ for $r_{\max} = 10^2$ km.

$\delta_\theta = 10^{-16}$), single precision (7 significant digits), and an intermediate precision of 9 significant digits. Note that with double precision the additional noise is negligible.

Note that in Theorem 3 the restriction about $r_{\max}$ is crucial. Namely, $\epsilon$-geo-indistinguishability does not hold for arbitrary distances for any finite $\epsilon$. Intuitively, this is because the step units of $\mathcal{W}$ (see Figure 4) become larger with the distance $r$ from $x_0$. The step units of $\mathcal{G}$, on the other hand, remain the same. When the steps in $\mathcal{W}$ become larger than those of $\mathcal{G}$, some $x$'s have an empty $R_{\mathcal{W}}(x)$. Therefore when $x$ is far away from $x_0$ its probability may or may not be 0, depending on the position of $x_0$ in $\mathcal{G}$, which means that geo-indistinguishability cannot be satisfied.

### *4.3. Truncation*

The Laplace mechanisms described in the previous sections have the potential to generate points everywhere in the plane, which causes several issues: first, digital applications have finite memory, hence these mechanisms are not implementable. Second, the discretized mechanism of Section 4.2 satisfies geo-indistinguishability only within a certain range, not on the full plane. Finally, in practical applications we are anyway interested in locations within a finite region (the earth itself is finite), hence it is desirable that the reported location lies within that region. For the above reasons, we propose a truncated variant of the discretized mechanism which generates points only within a specified region and fully satisfies geo-indistinguishability. The full mechanism (with discretization and truncation) is referred to as "Planar Laplace mechanism" and denoted by $PL_\epsilon$.

We assume a finite set $\mathcal{A} \subset \mathbb{R}^2$ of admissible locations, with diameter $\mathrm{diam}(\mathcal{A})$ (maximum distance between points in $\mathcal{A}$). This set is *fixed*, i.e. it does not depend on the actual location $x_0$. Our truncated mechanism $PL_\epsilon : \mathcal{A} \to \mathcal{P}(\mathcal{A} \cap \mathcal{G})$ works like the discretized Laplacian of the previous section, with the difference that the point generated in step (a) is remapped to the closest point in $\mathcal{A} \cap \mathcal{G}$. The complete mechanism is shown in Figure 6; note that step 1 assumes that $\mathrm{diam}(\mathcal{A}) < {}^{u}\!/_{\delta_\theta}$, otherwise no such $\epsilon'$ exists.

**Theorem 4.** *$PL_\epsilon$ satisfies $\epsilon$-geo-indistinguishability. namely*

$$K^T_{\epsilon'}(x)(z) \leq e^{\epsilon\, d(x,x')} K^T_{\epsilon'}(x')(z) \quad \textit{for every } x, x' \in \mathcal{A}$$

## 5. Enhancing LBSs with Privacy

In this section we present a case study of our privacy mechanism in the context of LBSs. In particular we show how to enhance LBS applications with privacy guarantees while still providing a high quality

**Input:** $x$ // *point to sanitize*
$\epsilon$ // *privacy parameter*
$u, v, \delta_\theta, \delta_r$ // *precision parameters*
$\mathcal{A}$ // *acceptable locations*
**Output:** Sanitized version $z$ of input $x$

1. $\epsilon' \leftarrow$ max $\epsilon'$ satisfying Thm 3 for $r_{\max} = \mathrm{diam}(\mathcal{A})$
2. draw $\theta$ unif. in $[0, 2\pi)$ // *draw angle*
3. draw $p$ unif. in $[0, 1)$, set $r \leftarrow C_{\epsilon'}^{-1}(p)$ // *draw radius*
4. $z \leftarrow x + \langle r\cos(\theta), r\sin(\theta)\rangle$ // *to cartesian, add vectors*
5. $z \leftarrow \mathrm{closest}(z, \mathcal{A})$ // *truncation*
6. **return** $z$

Fig. 6. The Planar Laplace mechanism $PL_\epsilon$

service to their users. We assume a simple client-server architecture where users communicate via a trusted mobile application (the client – typically installed in a smart-phone) with an unknown/untrusted LBS provider (the server – typically running on the cloud). Hence, in contrast to other solutions proposed in the literature, our approach does not rely on trusted third-party servers.

In the following we distinguish between *mildly-location-sensitive* and *highly-location-sensitive* LBS applications. The former category corresponds to LBS applications offering a service that does not heavily rely on the precision of the location information provided by the user. Examples of such applications are weather forecast applications and LBS applications for retrieval of certain kind of points of interest (POI), like gas stations or cinemas. Enhancing this kind of LBSs with geo-indistinguishability is relatively straightforward as it only requires to obfuscate the user's location using the Planar Laplace mechanism (Figure 6).

Our running example lies within the second category: For the user sitting at Café Les Deux Magots, information about restaurants nearby Champs Élysées is considerably less valuable than information about restaurants around his location. Enhancing highly-location-sensitive LBSs with privacy guarantees is more challenging. Our approach consists on implementing the following three steps:

1. Implement the Planar Laplace mechanism (Figure 6) on the client application in order to report to the LBS server the user's obfuscated location $z$ rather than his real location $x$.
2. Due to the fact that the information retrieved from the server is about POI nearby $z$, the area of POI information retrieval should be increased. In this way, if the user wishes to obtain information about POI within, say, 300 m of $x$, the client application should request information about POI within, say, 1 km of $z$. Figure 7 illustrates this situation. We will refer to the blue circle as *area of interest* (AOI) and to the grey circle as *area of retrieval* (AOR).
3. Finally, the client application should filter the retrieved POI information (depicted by the pins within the area of retrieval in Figure 7) in order to provide to the user with the desired information (depicted by pins within the user's area of interest in Figure 7).

Ideally, the AOI should always be fully contained in the AOR. Unfortunately, due to the probabilistic nature of our perturbation mechanism, this condition cannot be guaranteed (note that the AOR is centered on a randomly generated location that can be arbitrarily distant from the real location). It is also worth noting that the client application cannot dynamically adjust the radius of the AOR in order to ensure that

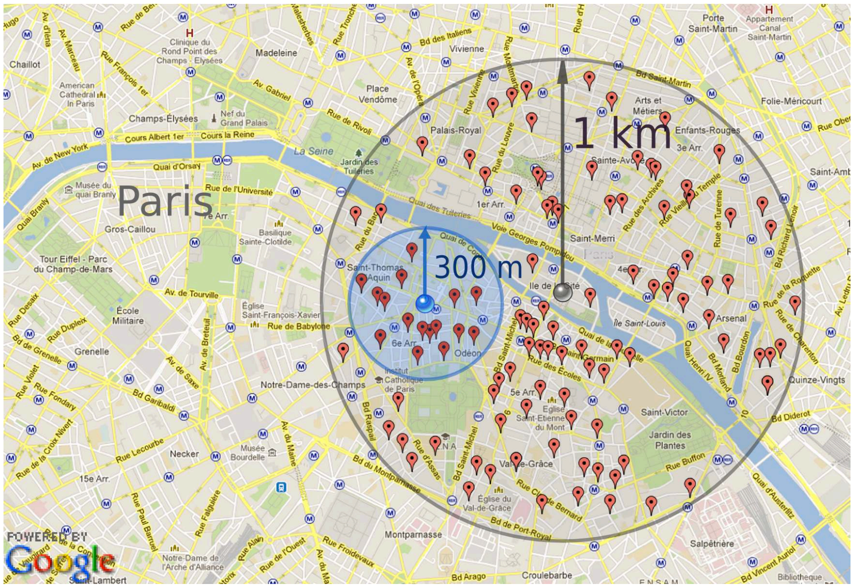



Fig. 7. AOI and AOR of $\mathbf{300}$ m and $\mathbf{1}$ km radius respectively.

it always contains the AOI as this approach would completely jeopardize the privacy guarantees: on the one hand, the size of the AOR would leak information about the user's real location and, on the other hand, the LBS provider would know with certainty that the user is located within the retrieval area. Thus, in order to provide geo-indistinguishability, the AOR has to be defined *independently* from the randomly generated location.

Since we cannot guarantee that the AOI is fully contained in the AOR, we introduce the notion of *accuracy*, which measures the probability of such event. In the following, we will refer to an LBS application in abstract terms, as characterized by a location perturbation mechanism $K$ and a fixed AOR radius. We use $rad_R$ and $rad_I$ to denote the radius of the AOR and the AOI, respectively, and $\mathcal{B}(x, r)$ to denote the circle with center $x$ and radius $r$.

### 5.1. *On the accuracy of LBS*s

Intuitively, an LBS application is $(c, rad_I)$-accurate if the probability of the AOI to be fully contained in the AOR is bounded from below by a *confidence factor* $c$. Formally:

**Definition 3** (LBS APPLICATION ACCURACY)**.** *An LBS application* $(K, rad_R)$ *is* $(c, rad_I)$*-accurate iff for all locations* $x$ *we have that* $\mathcal{B}(x, rad_I)$ *is fully contained in* $\mathcal{B}(K(x), rad_R)$ *with probability at least* $c$*.*

Given a privacy parameter $\epsilon$ and accuracy parameters $(c, rad_I)$, our goal is to obtain an LBS application $(K, rad_R)$ satisfying both $\epsilon$-geo-indistinguishability and $(c, rad_I)$-accuracy. As a perturbation mechanism, we use the Planar Laplace $PL_\epsilon$ (Figure 6), which satisfies $\epsilon$-geo-indistinguishability. As for $rad_R$, we aim at finding the minimum value validating the accuracy condition. Finding such minimum value is crucial to minimize the bandwidth overhead inherent to our proposal. In the following we will investigate how to achieve this goal by *statically* defining $rad_R$ as a function of the mechanism and the accuracy parameters $c$ and $rad_I$.

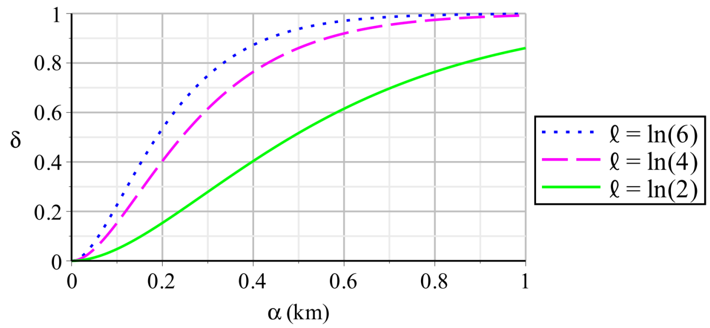



Fig. 8. $(\alpha, \delta)$-usefulness for $r = 0.2$ and various values of $\ell$.

For our purpose, it will be convenient to use the notion of ($\alpha$, $\delta$)-usefulness, which was introduced in [5]. A location perturbation mechanism $K$ is ($\alpha$, $\delta$)-*useful* if for every location $x$ the reported location $z = K(x)$ satisfies $d(x, z) \leq \alpha$ with probability at least $\delta$. In the case of the Planar Laplace, it is easy to see that, by definition, the $\alpha$ and $\delta$ values which express its usefulness are related by $C_\epsilon$ [4], the cdf of the Gamma distribution:

**Observation 1.** *For any $\alpha > 0$, $PL_\epsilon$ is ($\alpha$, $\delta$)-useful if $\alpha \leq C_\epsilon^{-1}(\delta)$.*

Figure 8 illustrates the ($\alpha$, $\delta$)-usefulness of $PL_\epsilon$ for $r = 0.2$ (as in our running example) and various values of $\ell$ (recall that $\ell = \epsilon\, r$). It follows from the figure that a mechanism providing the privacy guarantees specified in our running example ($\epsilon$-geo-indistinguishability, with $\ell = \ln(4)$ and $r = 0.2$) generates an approximate location $z$ falling within 1 km of the user's location $x$ with probability $0.992$, falling within $690$ meters with probability $0.95$, falling within $560$ meters with probability $0.9$, and falling within $390$ meters with probability $0.75$.

We now have all the necessary ingredients to determine the desired $rad_R$: By definition of usefulness, if $PL_\epsilon$ is ($\alpha$, $\delta$)-useful then the LBS application $(PL_\epsilon, rad_R)$ is ($\delta$, $rad_I$)-accurate if $\alpha \leq rad_R - rad_I$. The converse also holds if $\delta$ is maximal. By Observation 1, we then have:

**Proposition 1.** *The LBS application $(PL_\epsilon, rad_R)$ is ($c$, $rad_I$)-accurate if $rad_R \geq rad_I + C_\epsilon^{-1}(c)$.*

Therefore, it is sufficient to set $rad_R = rad_I + C_\epsilon^{-1}(c)$.

Coming back to our running example ($\epsilon = \ln(4)/0.2$ and $rad_I = 0.3$), taking a confidence factor $c$ of, say, $0.95$, leads to a $(0.69, 0.95)$-useful mechanism (because $C_\epsilon^{-1}(c) = 0.69$). Thus, $(PL_\epsilon, 0.99)$ is both $\ln(4)/0.2$-geo-indistinguishable and $(0.95, 0.3)$-accurate.

### 5.2. Bandwidth overhead analysis

As expressed by Proposition 1, in order to implement an LBS application enhanced with geo-indistinguishability and accuracy it suffices to use the Planar Laplace mechanism and retrieve POIs for an enlarged radius $rad_R$. For each query made from a location $x$, the application needs to (i) obtain $z = PL_\epsilon(x)$, (ii) retrieve POIs for AOR = $\mathcal{B}(z, rad_R)$, and (iii) filter the results from AOR to AOI (as explained in step 3 above). Such implementation is straightforward and computationally efficient for modern smart-phone devices. In addition, it provides great flexibility to application developer and/or users to

[4]For simplicity we assume that $\epsilon' = \epsilon$ (see Figure 6), since their difference is negligible under double precision.

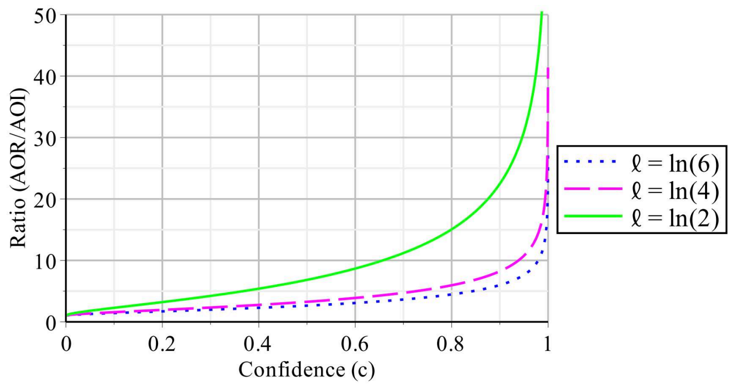



Fig. 9. AOR vs AOI ratio for various levels of privacy and accuracy (using fixed $r = 0.2$ and $rad_I = 0.3$).

specify their desired/allowed level of privacy and accuracy. This, however, comes at a cost: bandwidth overhead.

In the following we turn our attention to investigating the bandwidth overhead yielded by our approach. We will do so in two steps: first we investigate how the AOR size increases for different privacy and LBS-specific parameters, and then we investigate how such increase translates into bandwidth overhead.

Figure 9 depicts the overhead of the AOR versus the AOI (represented as their ratio) when varying the level of confidence ($c$) and privacy ($\ell$) and for fixed values $rad_I = 0.3$ and $r = 0.2$. The overhead increases slowly for levels of confidence up to $0.95$ (regardless of the level of privacy) and increases sharply thereafter, yielding to a worst case scenario of a about $50$ times increase for the combination of highest privacy ($\ell = \log(2)$) and highest confidence ($c = 0.99$).

In order to understand how the AOR increase translates into bandwidth overhead, we now investigate the density (in km$^2$) and size (in KB) of POIs by means of the Google Places API [1]. This API allows to retrieve POIs' information for a specific `location`, `radius` around the location, and POI's `type` (among many other optional parameters). For instance, the HTTPS request:

```
https://maps.googleapis.com/maps/api/place/nearbysearch/json?location=
48.85412,2.33316&radius=300&types=restaurant&key=myKey
```

returns information (in JSON format) including location, address, name, rating, and opening times for all restaurants up to 300 meters from the location $(48.85412, 2.33316)$ – which corresponds to the coordinates of Café Les Deux Magots in Paris.

We have used the APIs `nearbysearch` and `radarsearch` to calculate the average number of POIs per km$^2$ and the average size of POIs' information (in KB) respectively. We have considered two queries: restaurants in Paris, and restaurants in Buenos Aires. Our results show that there is an average of $137$ restaurants per km$^2$ in Paris and 22 in Buenos Aires, while the average size per POI is $0.84$ KB.

Combining this information with the AOR overhead depicted in Figure 9, we can derive the average bandwidth overhead for each query and various combinations of privacy and accuracy levels. For example, using the parameter combination of our running example (privacy level $\epsilon = \log(4)/0.2$, and accuracy level $c = 0.95$, $rad_I = 0.3$) we have a $10.7$ ratio for an average of 38 ($\backsimeq (137/1000^2) \times (300^2 \times \pi)$) restaurants in the AOI. Thus the estimated bandwidth overhead is $39 \times (10.7 - 1) \times 0.84$KB $\backsimeq 318$ KB.

Table 1 shows the bandwidth overhead for restaurants in Paris and Buenos Aires for the various combinations of privacy and accuracy levels. Looking at the worst case scenario, from a bandwidth overhead perspective, our combination of highest levels of privacy and accuracy (taking $\ell = \log(2)$ and $c = 0.99$)

with the query for restaurants in Paris (which yields to a large number of POIs – significantly larger than average) results in a significant bandwidth overhead (up to 1.7MB). Such overhead reduces sharply when decreasing the level of privacy (e.g., from $1.7$ MB to $557$ KB when using $\ell = \log(4)$ instead of $\ell = \log(2)$). For more standard queries yielding a lower number of POIs, in contrast, even the combination of highest privacy and accuracy levels results in a relatively insignificant bandwidth overhead.

| Restaurants in Paris | | **Accuracy** $rad_I = 0.3$ $c = 0.9$ | $c = 0.95$ | $c = 0.99$ |
|---|---|---|---|---|
| **Privacy** $r = 0.2$ | $\ell = \log(6)$ | 162 KB | 216 KB | 359 KB |
| | $\ell = \log(4)$ | 235 KB | 318 KB | 539 KB |
| | $\ell = \log(2)$ | 698 KB | 974 KB | 1.7 MB |

| Restaurants in Buenos Aires | | **Accuracy** $rad_I = 0.3$ $c = 0.9$ | $c = 0.95$ | $c = 0.99$ |
|---|---|---|---|---|
| **Privacy** $r = 0.2$ | $\ell = \log(6)$ | 26 KB | 34 KB | 54 KB |
| | $\ell = \log(4)$ | 38 KB | 51 KB | 86 KB |
| | $\ell = \log(2)$ | 112 KB | 156 KB | 279 KB |

Table 1

Bandwidth overhead for restaurants in Paris and in Buenos Aires for various levels of privacy and accuracy.

Concluding our bandwidth overhead analysis, we believe that the overhead necessary to enhance an LBS application with geo-indistinguishability guarantees is not prohibitive even for scenarios resulting in high bandwidth overhead (i.e., when combining very high privacy and accuracy levels with queries yielding a large number of POIs). Note that $1.7$MB is comparable to $35$ seconds of Youtube streaming or $80$ seconds of standard Facebook usage [2]. Nevertheless, for cases in which minimizing bandwidth consumption is paramount, we believe that trading bandwidth consumption for privacy (e.g., using $\ell = \log(4)$ or even $\ell = \log(6)$) is an acceptable solution.

## 6. Sanitizing datasets: US census case study

In this section we present a sanitation algorithm for datasets containing geographical information. We consider a realistic case study involving publicly available data developed by the U.S Census Bureau's Longitudinal Employer-Household Dynamics Program (LEHD). These data, called LEHD Origin-Destination Employment Statistics (LODES), are used by OnTheMap, a web-based interactive application developed by the US Census Bureau. The application enables, among other features, visualization of geographical information involving the residence and working location of US residents (e.g., distance from home to work location).

The LODES dataset includes information of the form $(hBlock, wBlock)$, where each pair represents a worker, the attribute $hBlock$ is the census block in which the worker lives, and $wBlock$ is the census block where the worker works. From this dataset it is possible to derive, by mapping home and work census blocks into their corresponding geographic centroids, a dataset with geographic information of the form ($hCoord$ , $wCoord$), where each of the coordinate pairs corresponds to a census block pair.

The Census Bureau uses a *synthetic data generation algorithm* [33,27] to sanitize the LODES dataset. Roughly speaking, the algorithm interprets the dataset as an histogram where each $(hBlock, wBlock)$ pair is represented by a histogram bucket, the synthetic data generation algorithm sanitizes data by modifying the counts of the histogram.

In the following we present a sanitizing algorithm for datasets with geographical information (e.g. the LODES dataset) that provides geo-indistinguishability guarantees under the assumption that the home

**Sanitizing Algorithm for a Dataset of Locations**

**Input:** $D : hCoord \times wCoord$ *// dataset to sanitize*
$\ell, r, u, v, \delta_r, \delta_\theta, \mathcal{A}$ *// same as in Figure 6*
**Output:** Sanitized version $D'$ of input $D$

1. $D' = \emptyset$; *// initializing output dataset*
2. $\epsilon = {}^{l}\!/\!{}_{r}$;
3. **for each** $(c_h, c_w) \in D$ **do**
4. $c'_h = NoisyPt(c_h, \epsilon, u, v, \delta_\theta, \delta_r, \mathcal{A})$;*// sanitized point*
5. $D' = D' \cup \{(c'_h, c_w)\}$; *// adding sanitized point*
6. **end-for**
7. **return** $D'$;

Fig. 10. Our sanitizing algorithm, based on data perturbation

census blocks values in the dataset are uncorrelated. Although this assumption weakens the privacy guarantees provided by geo-indistinguishability, we believe that due to the anonymizing techniques applied by the Census Bureau to the released data involving census participants' information and to the large number of ($hCoord$, $wCoord$) pairs within small areas contained in the dataset, a practical attack based on correlation of points is unlikely.

Our sanitizing algorithm, described in Figure 10, takes as input (1) a dataset $D$ to sanitize, (2) the privacy parameters $\ell$ and $r$ (see Section 3), and (3) the precision parameters $u$, $v$, $\delta_r$ and $\delta_\theta$, and the region $\mathcal{A}$. (see Section 4.2) and returns a sanitized counterpart of $D$. The algorithm is guaranteed to provide ${}^{\ell}\!/\!{}_{r}$-geo-indistinguishability to the home coordinates of all individuals in the dataset (see discussion on protecting multiple locations in Section 3.3).

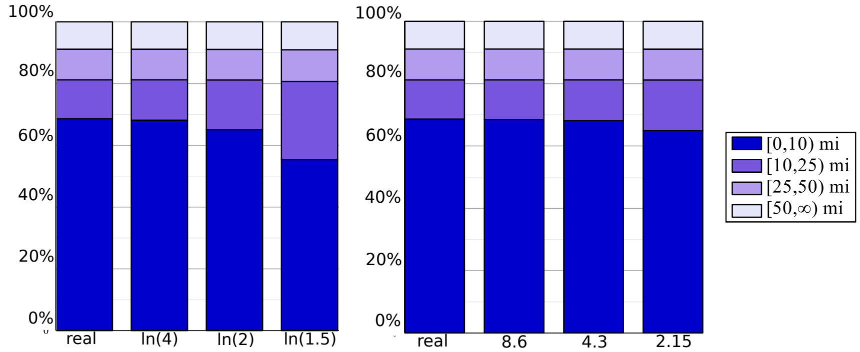
We note that, in contrast to the approach used by the Census Bureau based on histogram's count perturbation, our algorithm modifies the geographical data itself (residence coordinates in this case). Therefore, our algorithm works at a more refined level than the synthetic data generation algorithm used by the Census Bureau; a less refined dataset can be easily obtained however – by just remapping each $(hCoord, wCoord)$ pair produced by our algorithm to its corresponding census block representation.

### *6.1. Experiments on the LODES dataset*

In order to evaluate the accuracy of the sanitized dataset generated by our algorithm (and thus of our algorithm as a data sanitizer) we implemented our perturbation mechanism and conducted a series of experiments focusing on the "home-work commute distance" analysis provided by the OnTheMap application. This analysis provides, for a given area (specified as, say, state or county code), a histogram classifying the individuals in the dataset residing in the given area according to the distance between their residence location and their work location. The generated histogram contains four buckets representing different ranges of distance: (1) from zero to ten miles, (2) from ten to twenty five miles, (3) from twenty five to fifty miles, and (4) more than fifty miles.[5].

We have chosen the San Francisco (SF) County as residence area for our experimental analysis. Additionally, we restrict the work location of individuals residing in the San Francisco county to the state of California. The total number of individuals satisfying these conditions amounts to 374.390. All experi-

[5]Here we choose miles as unit of measure, in order to compare our results with the literature and with online tools about the LODES dataset.




(a) $r = 1.22$ (b) $r = 3.86$

Fig. 11. Home-work commute distance for various levels $\ell$.

ments have been carried on using version 6.0 of the LODES dataset. In addition, the mapping from census blocks to their corresponding centroids has been done using the 2011 TIGER census block shapefile information provided by the Census Bureau.

We now proceed to compare the LODES dataset – seen as a histogram – with several sanitized versions of it generated by our algorithm. Figure 11 (a) depicts how the geographical information degrades when fixing $r$ to 1.22 miles (so to ensure geo-indistinguishability within $10\%$ of the land area of the SF County) and varying $\ell$. The precision parameters were chosen as follows: $u = 10^{-3}$ miles, $\mathcal{A}$'s diameter was set to $10^4$ miles, and the standard double precision values for $\delta_r$ and $\delta_\theta$ (for the corresponding ranges).

We have also conducted experiments varying $r$ and fixing $\ell$. For instance, if we want to provide geo-indistinguishability for $5\%$, $10\%$, and $25\%$ of the land area of the SF county (approx. $46.87$ mi$^2$), we can set $r = 0.86$, $1.22$, and $1.93$ miles, respectively. Then by taking $\ell = \ln(2)$ we get an histogram very similar to the previous one. This is not surprising as the noise generated by our algorithm depends only on the ratios $\ell/r$, which are similar for the values above.

As shown in Figure 11 (a), our algorithm has little effect on the bucket counts corresponding to mid/long distance commutes: over twenty five miles the counts of the sanitized dataset are almost identical to those of the input dataset – even for the higher degrees of privacy. For short commutes on the other hand, the increase in privacy degrades the accuracy of the sanitized dataset: several of the commutes that fall in the 0-to-10-miles bucket in the original data fall instead in the 10-to-25-miles bucket in the sanitized data.

After analyzing the accuracy of the sanitized datasets produced by our algorithm for several levels of privacy, we proceed to compare our approach with the one followed by the Census Bureau to sanitize the LODES dataset. Such comparison is unfortunately not straightforward; on the one hand, the approaches provide different privacy guarantees (see discussion below) and, on the other hand, the Census Bureau is not able to provide us with a (sanitized) dataset sample produced by their algorithm (which would allow us to compare both approaches in terms of accuracy) as this might compromise the protection of the real data.

The algorithm used by the Census Bureau satisfies a notion of privacy called $(\epsilon, \delta)$-probabilistic differential privacy (which is a relaxation of standard definition of differential privacy) that provides $\epsilon$-differential privacy with probability at least $1 - \delta$ [27]. In particular, their algorithm satisfies $(8.6, 0.00001)$ -probabilistic differential privacy. This level of privacy could be compared to geo-indistinguishability for

$\ell = 8.6$ and $r = 3.86$, which corresponds to providing protection in an area of the size of the SF County. Figure 11 (b) presents the results of our algorithm for such level of privacy and also for higher levels.

It becomes clear that, by allowing high values for $\ell$ ($\ell = 8.6 = \ln(5432)$, $\ell = 4.3 = \ln(74)$, and $\ell = 2.15 = \ln(9)$) it is possible to provide privacy in large areas without significantly diminishing the quality of the sanitized dataset.

## 7. Comparison with other methods

In this section we compare the performance of our mechanism with that of other ones proposed in the literature. Of course it is not interesting to make a comparison in terms of geo-indistinguishability, since other mechanisms usually do not satisfy this property.

We consider, instead, the (rather natural) Bayesian notion of privacy that measures the expected error of the adversary, presented in Section 2.1. We also consider the trade-off with respect to the quality loss (measured as the expected distance between the real location and the reported result), and also with respect to the notion of accuracy illustrated in the previous section.

The mechanisms that we compare with ours are:

1. The obfuscation mechanism presented in [37]. This mechanism works on discrete locations, called *regions*, and, like ours, it reports a location (region) selected randomly according to a probability distribution that depends on the user's location. The distributions are generated automatically by a tool which is designed to produce an OptPriv mechanism (see Section 2.1), that is, a mechanism that provides optimal privacy for a given utility and a given adversary (i.e., a given prior, representing the side knowledge of the adversary). It is important to note that in presence of a different adversary the optimality is not guaranteed. This dependency on the prior is a key difference with respect to our approach, which abstracts from the adversary's side information.
2. A simple cloaking mechanism. In this approach, the area of interest is assumed to be partitioned in *zones*, whose size depends on the level of privacy we want yo achieve. The mechanism then reports the zone in which the exact location is situated. This method satisfies $k$-anonymity where $k$ is the number of locations within each zone.

In both cases we need to divide the considered area into a finite number of regions, representing the possible locations. We consider for simplicity a grid, and, more precisely, a $9 \times 9$ grid consisting of $81$ square regions of 100 m of side length. In addition, for the cloaking method, we overlay a grid of $3 \times 3 = 9$ zones. Figure 12 illustrates the setting: the regions are the small squares with black borders. In the cloaking method, the zones are the larger squares with blue borders. For instance, any point situated in one of the regions $1, 2, 3, 10, 11, 12, 19, 20$ or $21$, would be reported as zone $1$. We assume that each zone is represented by the central region. Hence, in the above example, the reported region would be $11$.

*Measuring privacy and utility* As already stated, we will use the metrics for location privacy and for the quality loss proposed in [37] and described in Section 2.1. The expected error of the advesary is used to measure the privacy offered by a mechanism, and it is formally defined as follows:

$$\textsc{AdvError}(K, \pi) = \min_H \sum_{x,\hat{x}} \pi(x)(KH)(x)(\hat{x})d_2(x, \hat{x})$$

where $\pi$ is the prior distribution over the locations, $K(x)(z)$ gives the probability that the real location $x$ is reported by the mechanism as $z$, $H$ is called a remapping, where $H(z)(\hat{x})$ represents the probability that the reported region $z$ is remapped into $\hat{x}$ and $d_2$ is the Euclidean distance between locations. As for

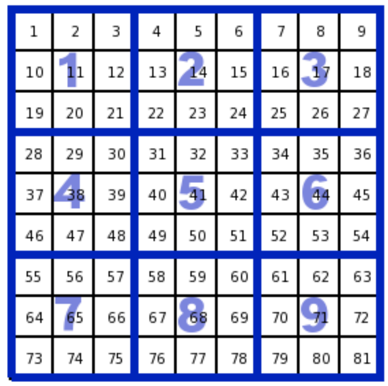


Fig. 12. The division of the map into regions and zones.

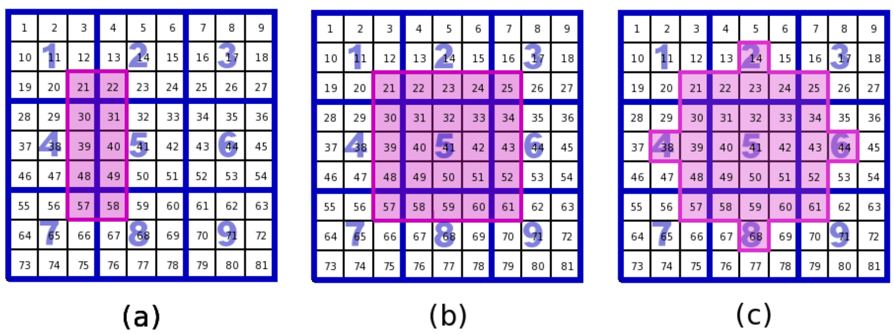


Fig. 13. Priors considered for the experiments.

the utility, we quantify its opposite, the *Quality Loss* (QL), in terms of the expected distance between the reported location and the user's exact location:

$$\mathrm{QL}(K,\pi) = \textstyle\sum_{x,z} \pi(x)K(x)(z)d_2(x,z)$$

where $\pi$ and $K$ are as above. We note that actually the definitions of ADVERROR and QL presented in Section 2.1 depend respectively on the metric used by the adversary to measure the success of her guessing, and on the one used by the user to measure the quality of the obtained results. In this section we assume that both these metrics are equal to the Euclidean distance $d_2$, and therefore for simplicity we omit them from the list of arguments.

Recall that for the optimal mechanism in [37] QL and ADVERROR coincide (when the mechanism is used in presence of the same adversary for which it has been designed), i.e. the adversary does not need to make any remapping.

*Comparing privacy for a given utility* In order to compare the three mechanisms, we set the parameters of each mechanism in such a way that the QL is the same for all of them, and we compare their privacy in terms of ADVERROR. As already noted, for the OPTPRIV mechanism generated by the approach of [37] QL and ADVERROR coincide, i.e. the optimal remapping is the identity, when the mechanism is used in presence of the same adversary for which it has been designed. It turns out that, when the adversary's

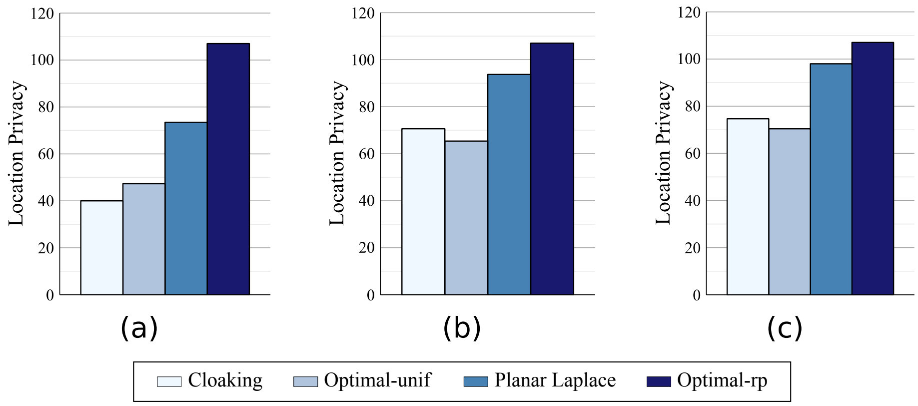



Fig. 14. Location Privacy, in terms of ADVERROR, for QL $= 107.03$ m, for the four mentioned mechanisms, under priors (a) $\pi_1$, (b) $\pi_2$ and (c) $\pi_3$.

prior is the uniform one, QL and ADVERROR coincide also for the Planar Laplace mechanism and for the cloaking one.

We note that for the cloaking mechanism the QL is fixed and it is $107.03$ m, since there is no randomization involved in the generation of the reported location. Therefore, in our experiments we fix the value of QL to be that one for all the mechanisms. We find that in order to obtain such QL for the Planar Laplace mechanism we need to set $\epsilon = 0.0162$ (the difference with $\epsilon'$ in this case is negligible). The OPTPRIV mechanism of [37] is generated by using the tool explained in the same paper.

Figure 13 illustrates the priors $\pi_1$, $\pi_2$ and $\pi_3$ that we consider here: in each case, the probability distribution is accumulated in the regions in the purple area, and distributed uniformly over them. Note that it is not interesting to consider the uniform distribution over the whole map, since, as explained before, on that prior all the mechanisms under consideration give the same result.

Figure 14 illustrates the results we obtain when comparing privacy in terms of ADVERROR, where (a), (b) and (c) correspond to the priors $\pi_1$, $\pi_2$ and $\pi_3$ in Figure 13 respectively. The optimal mechanism is considered in two instances: the one designed exactly for the prior for which it is used ($\text{OPTPRIV}_{\pi_i}$, for $i \in \{1, 2, 3\}$), and the one designed for the uniform distribution on all the map ($\text{OPTPRIV}_u$, which is not necessarily optimal for the priors considered here). As we can see, the Planar Laplace mechanism offers the best location privacy among the mechanisms which do not depend on the prior, or, as in the case of $\text{OPTPRIV}_u$, are designed with a fixed prior. When the prior has a more circular symmetry the performance approaches the one of the corresponding $\text{OPTPRIV}_{\pi_i}$ mechanism (which offers the optimal privacy for that prior).

*Comparing privacy for a given accuracy* The QL metric used above is a reasonable metric, but it does not cover all natural notions of utility. In particular, in the case of LBSs, an important criterion to take into account is the additional bandwidth usage. Therefore, we make now a comparison using the notion of accuracy, which, as explained in previous section, provides a good criterion to evaluate the performance in terms of bandwidth. Unfortunately we cannot compare our mechanism to the one of [37] under this criterion, because the construction of the latter is tied to the a fixed value of QL. Hence, we only compare our mechanism with the cloaking one.

We recall that an LBS application $(K, rad_R)$ is $(c, rad_I)$-accurate if for every location $x$ the probability that the area of interest (AOI) is fully contained in the area of retrieval (AOR) is at least $c$. We need to fix $rad_I$ (the radius of the AOI), $rad_R$ (the radius of the AOR), and $c$ so that the condition of accuracy is

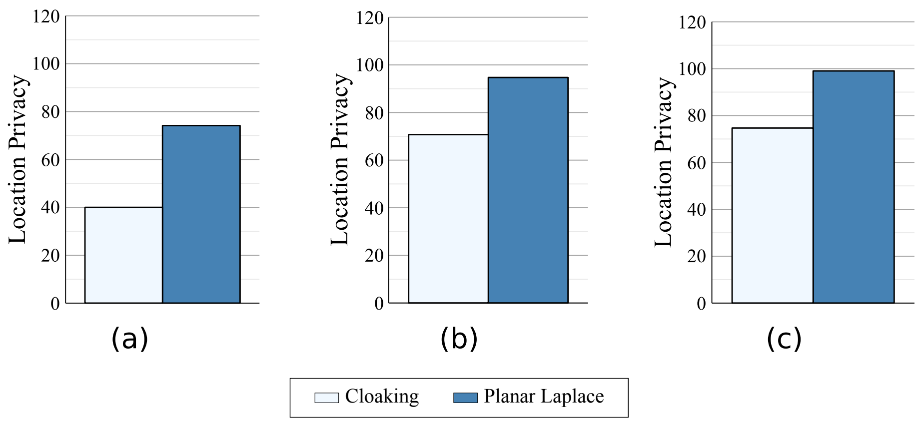



Fig. 15. Location Privacy for $rad_R = (\sqrt{2} \cdot 150 + 200)$ m and $c = 0.99$.

satisfied for both methods, and then compute the respective ADVERROR. Let us fix $rad_I = 200$ m, and let us choose a large confidence factor, say, $c = 0.99$. As for $rad_R$, it will be determined by the cloaking method.

Since the cloaking mechanism is deterministic, in order for the condition to be satisfied the AOR for a given location $x$ must extend around the zone of $x$ by at least $rad_I$, In fact, $x$ could be in the border of the zone. Given that the cloaking method reports the center of the zone, and that the distance between the center and the border (which is equal to the distance between the center and any of the corners) is $\sqrt{2} \cdot 150$ m, we derive that $rad_R$ must be at least $(200 + \sqrt{2} \cdot 150)$ m. Note that in the case of this method the accuracy is independent from the value of $c$. It only depends on the difference between $rad_R$ and $rad_I$, which in turns depends on the length $s$ of the side of the region: if the difference is at least $\sqrt{2} \cdot s/2$, then the condition is satisfied (for every possible $x$) with probability 1. Otherwise, there will be some $x$ for which the condition is not satisfied (i.e., it is satisfied with probability 0).

In the case of our method, on the other hand, the accuracy condition depends on $c$ and on $\epsilon$. More precisely, as we have seen in previous section, the condition is satisfied if and only if $C_\epsilon^{-1}(c) \leq rad_R - rad_I$. Therefore, for fixed $c$, the maximum $\epsilon$ only depends on the difference between $rad_R$ and $rad_I$ and is determined by the equation $C_\epsilon^{-1}(c) = rad_R - rad_I$. For the above values of $rad_I$, $rad_R$, and $c$, it turns out that $\epsilon = 0.016$.

We can now compare the ADVERROR of the two mechanisms with respect to the three priors above. Figure 15 illustrates the results. As we can see, our mechanism outperforms the cloaking mechanism in all the three cases.

For different values of $rad_I$ the situation does not change: as explained above, the cloaking method always forces $rad_R$ to be larger than $rad_I$ by (at least) $\sqrt{2} \cdot 150$ m, and $\epsilon$ only depends on this value. For smaller values of $c$, on the contrary, the situation changes, and becomes more favorable for our method. In fact, as argued above, the situation remains the same for the cloaking method (since its accuracy does not depend on $c$), while $\epsilon$ decreases (and consequently ADVERROR increases) as $c$ decreases. In fact, for a fixed $r = rad_R - rad_I$, we have $\epsilon = C_r^{-1}(c)$. This follows from $r = C_\epsilon^{-1}(c)$ and from the fact that $r$ and $\epsilon$, in the expression that defines $C_\epsilon(r)$, are interchangeable.

## 8. Related Work

Much of the related work has been already discussed in Section 2.2, here we only mention the works that were not reported there. There are excellent works and surveys [39,25,35] that summarize the different threats, methods, and guarantees in the context of location privacy.

LISA [8] provides location privacy by preventing an attacker from relating any particular point of interest (POI) to the user's location. That way, the attacker cannot infer which POI the user will visit next. The privacy metric used in this work is $m$-*unobservability*. The method achieves $m$-unobservability if, with high probability, the attacker cannot relate the estimated location to at least $m$ different POIs in the proximity.

SpaceTwist [41] reports a fake location (called the "anchor") and queries the geolocation system server incrementally for the nearest neighbors of this fake location until the $k$-nearest neighbors of the real location are obtained.

In a recent paper [29] it has been shown that, due to finite precision and rounding effects of floating-point operations, the standard implementations of the Laplacian mechanism result in an irregular distribution which causes the loss of the property of differential privacy. In [17] the study has been extended to the planar Laplacian, and to any kind of finite-precision semantics. The same paper proposes a solutions for the truncated version of the planar laplacian, based on a snapping meccanism, which maintains the level of privacy at the cost of introducing an additional amount of noise.

## 9. Conclusion

In this paper we have presented a framework for achieving privacy in location-based applications, taking into account the desired level of protection as well as the side-information that the attacker might have. The core of our proposal is a new notion of privacy, that we call geo-indistinguishability, and a method, based on a bivariate version of the Laplace function, to perturbate the actual location. We have put a strong emphasis in the formal treatment of the privacy guarantees, both in giving a rigorous definition of geo-indistinguishability, and in providing a mathematical proof that our method satisfies such property. We also have shown how geo-indistinguishability relates to the popular notion of differential privacy. Finally, we have illustrated the applicability of our method on a POI-retrieval service, and we have compared it with other mechanisms in the literature, showing that it outperforms those which do not depend on the prior.

## 10. Acknowledgements

This work was partially supported by the European Union 7th FP under the grant agreement no. 295261 (MEALS), by the projects ANR-11-IS02-0002 LOCALI and ANR-12-IS02-001 PACE, and by the INRIA Large Scale Initiative CAPPRIS. The work of Miguel E. Andrés was supported by a QUALCOMM grant. The work of Nicolás E. Bordenabe was partially funded by the French Defense Procurement Agency (DGA) by a PhD grant.

## References

[1] Google Places API. `https://developers.google.com/places/documentation/`.

[2] Vodafone Mobile data usage Stats. `http://www.vodafone.ie/internet-broadband/internet-on-your-mobile/usage/`.

[3] C. A. Ardagna, M. Cremonini, E. Damiani, S. D. C. di Vimercati, and P. Samarati. Location privacy protection through obfuscation-based techniques. In S. Barker and G.-J. Ahn, editors, *Proc. of the 21st Annual IFIP WG 11.3 Working Conference on Data and Applications Security (DAS)*, volume 4602 of *Lecture Notes in Computer Science*, pages 47–60. Springer, 2007.

[4] B. Bamba, L. Liu, P. Pesti, and T. Wang. Supporting anonymous location queries in mobile environments with privacygrid. In *Proc. of the 17th International Conference on World Wide Web (WWW)*, pages 237–246. ACM, 2008.

[5] A. Blum, K. Ligett, and A. Roth. A learning theory approach to non-interactive database privacy. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC)*, pages 609–618. ACM, 2008.

[6] K. Chatzikokolakis, M. E. Andrés, N. E. Bordenabe, and C. Palamidessi. Broadening the scope of Differential Privacy using metrics. In E. De Cristofaro and M. Wright, editors, *Proceedings of the 13th International Symposium on Privacy Enhancing Technologies (PETS 2013)*, volume 7981 of *Lecture Notes in Computer Science*, pages 82–102. Springer, 2013.

[7] K. Chatzikokolakis, C. Palamidessi, and M. Stronati. A predictive differentially-private mechanism for mobility traces. *CoRR*, abs/1311.4008, 2013.

[8] Z. Chen. *Energy-efficient Information Collection and Dissemination in Wireless Sensor Networks.* PhD thesis, University of Michigan, 2009.

[9] R. Cheng, Y. Zhang, E. Bertino, and S. Prabhakar. Preserving user location privacy in mobile data management infrastructures. In G. Danezis and P. Golle, editors, *Proceedings of the 6th International Workshop on Privacy Enhancing Technologies (PET)*, volume 4258 of *Lecture Notes in Computer Science*, pages 393–412. Springer, 2006.

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

[11] J. E. Dobson and P. F. Fisher. Geoslavery. *Technology and Society Magazine, IEEE*, 22(1):47–52, 2003.

[12] M. Duckham and L. Kulik. A formal model of obfuscation and negotiation for location privacy. In *Proc. of the Third International Conference on Pervasive Computing (PERVASIVE)*, volume 3468 of *Lecture Notes in Computer Science*, pages 152–170. Springer, 2005.

[13] C. Dwork. Differential privacy. In M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, editors, *33rd International Colloquium on Automata, Languages and Programming (ICALP 2006)*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2006.

[14] C. Dwork. A firm foundation for private data analysis. *Communications of the ACM*, 54(1):86–96, 2011.

[15] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. S. Zemel. Fairness through awareness. In S. Goldwasser, editor, *Proceedings of the Third Innovations in Theoretical Computer Science (ITCS) conference*, pages 214–226. ACM, 2012.

[16] C. Dwork, F. Mcsherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In S. Halevi and T. Rabin, editors, *In Proceedings of the Third Theory of Cryptography Conference (TCC)*, volume 3876 of *Lecture Notes in Computer Science*, pages 265–284. Springer, 2006.

[17] I. Gazeau, D. Miller, and C. Palamidessi. Preserving differential privacy under finite-precision semantics. In L. Bortolussi and H. Wiklicky, editors, *Proceedings of the Eleventh Workshop on Quantitative Aspects of Programming Languages (QAPL 2013)*, volume 117 of *Electronic Proceedings in Theoretical Computer Science*, pages 1–18. Open Publishing Association, 2013.

[18] B. Gedik and L. Liu. Location privacy in mobile systems: A personalized anonymization model. In *Proc. of the 25th International Conference on Distributed Computing Systems (ICDCS)*, pages 620–629. IEEE Computer Society, 2005.

[19] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan. Private queries in location based services: anonymizers are not necessary. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 121–132. ACM, 2008.

[20] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proc. of the First International Conference on Mobile Systems, Applications, and Services (MobiSys)*. USENIX, 2003.

[21] S.-S. Ho and S. Ruan. Differential privacy for location pattern mining. In *Proceedings of the 4th ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS (SPRINGL)*, pages 17–24. ACM, 2011.

[22] B. Hoh and M. Gruteser. Protecting location privacy through path confusion. In *Proc. of SecureComm*, pages 194–205. IEEE, 2005.

[23] A. Khoshgozaran and C. Shahabi. Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy. In *Proc. of the 10th International Symposium on the Advances in Spatial and Temporal Databases (SSTD)*, volume 4605 of *Lecture Notes in Computer Science*, pages 239–257. Springer, 2007.

[24] H. Kido, Y. Yanagisawa, and T. Satoh. Protection of location privacy using dummies for location-based services. In *Proc. of ICDE Workshops*, page 1248, 2005.

[25] J. Krumm. A survey of computational location privacy. *Personal and Ubiquitous Computing*, 13(6):391–399, 2009.

[26] K. Lange and J. S. Sinsheimer. Normal/independent distributions and their applications in robust regression. *Journal of Computational and Graphical Statistics*, 2(2):175–198, 1993.

[27] A. Machanavajjhala, D. Kifer, J. M. Abowd, J. Gehrke, and L. Vilhuber. Privacy: Theory meets practice on the map. In G. Alonso, J. A. Blakeley, and A. L. P. Chen, editors, *Proceedings of the 24th International Conference on Data Engineering, ICDE 2008, April 7-12, 2008, Cancún, México*, pages 277–286. IEEE, 2008.
[28] Pew Internet. Smartphone Ownership, 2013.
[29] I. Mironov. On significance of the least significant bits for differential privacy. In T. Yu, G. Danezis, and V. D. Gligor, editors, *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, pages 650–661. ACM, 2012.
[30] M. F. Mokbel, C.-Y. Chow, and W. G. Aref. The new casper: Query processing for location services without compromising privacy. In U. Dayal, K.-Y. Whang, D. B. Lomet, G. Alonso, G. M. Lohman, M. L. Kersten, S. K. Cha, and Y.-K. Kim, editors, *Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB)*, pages 763–774. ACM, 2006.
[31] J. Reed and B. C. Pierce. Distance makes the types grow stronger: a calculus for differential privacy. In P. Hudak and S. Weirich, editors, *Proceeding of the 15th ACM SIGPLAN International Conference on Functional Programming (ICFP)*, pages 157–168, Baltimore, Maryland, USA, September 27-29 2010. ACM.
[32] A. Roth and T. Roughgarden. Interactive privacy via the median mechanism. In *Proc. of the 42nd ACM Symposium on Theory of Computing (STOC)*, pages 765–774, 2010.
[33] D. B. Rubin. Discussion: Statistical disclosure limitation. *Journal of Official Statistics*, 9(2):461–468, 1993.
[34] P. Shankar, V. Ganapathy, and L. Iftode. Privately querying location-based services with SybilQuery. In S. Helal, H. Gellersen, and S. Consolvo, editors, *Proc. of the 11th International Conference on Ubiquitous Computing (UbiComp)*, pages 31–40. ACM, 2009.
[35] K. G. Shin, X. Ju, Z. Chen, and X. Hu. Privacy protection for users of location-based services. *IEEE Wireless Commun*, 19(2):30–39, 2012.
[36] R. Shokri, G. Theodorakopoulos, J.-Y. L. Boudec, and J.-P. Hubaux. Quantifying location privacy. In *IEEE Symposium on Security and Privacy*, pages 247–262. IEEE Computer Society, 2011.
[37] R. Shokri, G. Theodorakopoulos, C. Troncoso, J.-P. Hubaux, and J.-Y. L. Boudec. Protecting location privacy: optimal strategy against localization attacks. In T. Yu, G. Danezis, and V. D. Gligor, editors, *Proceedings of the 19th ACM Conference on Computer and Communications Security (CCS 2012)*, pages 617–627. ACM, 2012.
[38] R. Shokri, C. Troncoso, C. Diaz, J. Freudiger, and J.-P. Hubaux. Unraveling an old cloak: k-anonymity for location privacy. In *Proceedings of the 9th annual ACM Workshop on Privacy in the Electronic Society (WPES 2010)*, pages 115–118 115–118 115–118, 2010.
[39] M. Terrovitis. Privacy preservation in the dissemination of location data. *SIGKDD Explorations*, 13(1):6–18, 2011.
[40] M. Xue, P. Kalnis, and H. Pung. Location diversity: Enhanced privacy protection in location based services. In *Proc. of the 4th International Symposium on Location and Context Awareness (LoCA)*, volume 5561 of *Lecture Notes in Computer Science*, pages 70–87. Springer, 2009.
[41] M. L. Yiu, C. S. Jensen, X. Huang, and H. Lu. Spacetwist: Managing the trade-offs among location privacy, query performance, and query accuracy in mobile services. In G. Alonso, J. A. Blakeley, and A. L. P. Chen, editors, *Proceedings of the 24th International Conference on Data Engineering (ICDE)*, pages 366–375. IEEE, 2008.

## Appendix

In this appendix we provide the technical details that have been omitted from the main body of the paper.

**Theorem 3.** *Assume $r_{\max} < {}^{u}/_{\delta_\theta}$, and let $q = {}^{u}/_{r_{\max}\delta_\theta}$. Let $\epsilon, \epsilon' \in \mathbb{R}^+$ such that*

$$\epsilon' + \frac{1}{u} \ln \frac{q + 2\, e^{\epsilon' u}}{q - 2\, e^{\epsilon' u}} \leq \epsilon$$

*Then $K_{\epsilon'}$ provides $\epsilon$-geo-indistinguishability within the range of $r_{\max}$. Namely, if $d(x,z), d(x',z) \leq r_{\max}$ then:*

$$K_{\epsilon'}(x)(z) \leq e^{\epsilon\, d(x,x')} K_{\epsilon'}(x')(z).$$

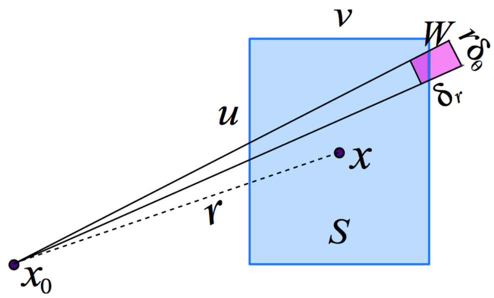



Fig. 16. Bounding the probability of $x$ in the discrete Laplacian.

*Proof.* The case in which $x_0 = x_0'$ is trivial. We consider therefore only the case in which $x_0 \neq x_0'$. Note that in this case $d(x_0, x_0') \geq u$. We proceed by determining an upper bound on $K_{\epsilon'}(x_0)(x)$ and a lover bound on $K_{\epsilon'}(x_0')(x)$ for generic $x_0$, $x_0'$ and $x$ such that $d(x_0, x), d(x_0', x) \leq r_{\max}$. Let $S$ be the set of points for which $x$ is the closest point in $\mathcal{G}$, namely:

$$S = R(x) = \{y \in \mathbb{R}^2 \,|\, \forall x' \in \mathcal{G}.\ d(y, x') \leq d(y, x')\}$$

Ideally, the points remapped in $x$ would be exactly those in $S$. However, due to the finite precision of the machine, the points actually remapped in $x$ are those of $R_{\mathcal{W}}(x)$ (see Section 4.2). Hence the probability of $x$ is that of $S$ plus or minus the small rectangles[6] $W$ of size $\delta_r \times r\,\delta_\theta$ at the border of $S$, where $r = d(x_0, x)$, see Figure 16. Let us denote by $S_W$ the total area of these small rectangles $W$ on one of the sides of $S$. Since $d(x_0, x) \leq r_{\max} < {}^{u}/_{\delta_\theta}$, and $\delta_r < r_{\max}\delta_\theta$, we have that $S_W$ is less than ${}^{1}/_{q}$ of the area of $S$, where $q = {}^{u}/_{r_{\max}\delta_\theta}$. The probability density on this area differs at most by a factor $e^{\epsilon' u}$ from that of the other points in $S$. Finally, note that on two sides of $S$ the rectangles $W$ contribute positively to $K_{\epsilon'}(x_0)(x)$, while on two sides they contribute negatively. Summarizing, we have:

$$K_{\epsilon'}(x_0)(x) \;\leq\; (1 + \frac{2\,e^{\epsilon' u}}{q}) \int_S D_{\epsilon'}(x_0)(x_1) ds \tag{3}$$

and

$$(1 - \frac{2\,e^{\epsilon' u}}{q}) \int_S D_{\epsilon'}(x_0')(x_1) ds \;\leq\; K_{\epsilon'}(x_0')(x) \tag{4}$$

Observe now that

$$\frac{D_{\epsilon'}(x_0)(x_1)}{D_{\epsilon'}(x_0')(x_1)} = e^{-\epsilon'(d(x_0,x_1) - d(x_0',x_1))}$$

By triangular inequality we obtain

$$D_{\epsilon'}(x_0)(x_1) \leq e^{\epsilon'\, d(x_0, x_0')} D_{\epsilon'}(x_0')(x_1)$$

[6] $W$ is actually a fragment of a circular crown, but since $\delta_\theta$ is very small, it approximates a rectangle. Also, the side of $W$ is not exactly $r\,\delta_\theta$, it is a number in the interval $[(r - {}^{u}/_{\sqrt{2}})\,\delta_\theta, (r + {}^{u}/_{\sqrt{2}})\,\delta_\theta]$. However ${}^{u}/_{\sqrt{2}}\,\delta_\theta$ is very small with respect to the other quantities involved, hence we consider negligible this difference.

from which we derive

$$\int_S D_{\epsilon'}(x_0)(x_1)ds \leq e^{\epsilon'\, d(x_0,x'_0)} \int_S D_{\epsilon'}(x'_0)(x_1)ds \tag{5}$$

from which, using (3), (5), and (4), we obtain

$$K_{\epsilon'}(x_0)(x) \leq e^{\epsilon'\, d(x_0,x'_0)}\, K_{\epsilon'}(x'_0)(x)\, \frac{q + 2\, e^{\epsilon' u}}{q - 2\, e^{\epsilon' u}} \tag{6}$$

Assume now that

$$\epsilon' + \frac{1}{u} \ln \frac{q + 2\, e^{\epsilon' u}}{q - 2\, e^{\epsilon' u}} \leq \epsilon$$

Since we are assuming $d(x_0, x'_0) \geq u$, we derive:

$$e^{\epsilon'\, d(x_0,x'_0)}\, \frac{q + 2\, e^{\epsilon' u}}{q - 2\, e^{\epsilon' u}} \leq e^{\epsilon\, d(x_0,x'_0)} \tag{7}$$

Finally, from (6) and (7), we conclude. □

**Theorem 4.** *$PL_\epsilon$ satisfies $\epsilon$-geo-indistinguishability. namely*

$$K^T_{\epsilon'}(x)(z) \leq e^{\epsilon\, d(x,x')} K^T_{\epsilon'}(x')(z) \quad \textit{for every } x, x' \in \mathcal{A}$$

*Proof.* The proof proceeds like the one for Theorem 3, except when $R(x)$ is on the border of $\mathcal{A}$. In this latter case, the probability on $x$ is given not only by the probability on $R(x)$ (plus or minus the small rectangles $W$ – see the proof of Theorem 3), but also by the probability of the part $C$ of the cone determined by $o$, $R(x)$, and lying outside $\mathcal{A}$ (see Figure 17). Following a similar reasoning as in the proof of Theorem 3 we get

$$K^T_{\epsilon'}(x_0)(x) \;\leq\; (1 + \frac{2\, e^{\epsilon' u}}{q}) \int_{S \cup C} D_{\epsilon'}(x_0)(x_1)ds$$

and

$$(1 - \frac{2\, e^{\epsilon' u}}{q}) \int_{S \cup C} D_{\epsilon'}(x'_0)(x_1)ds \;\leq\; K^T_{\epsilon'}(x'_0)(x)$$

The rest follows as in the proof of Theorem 3. □

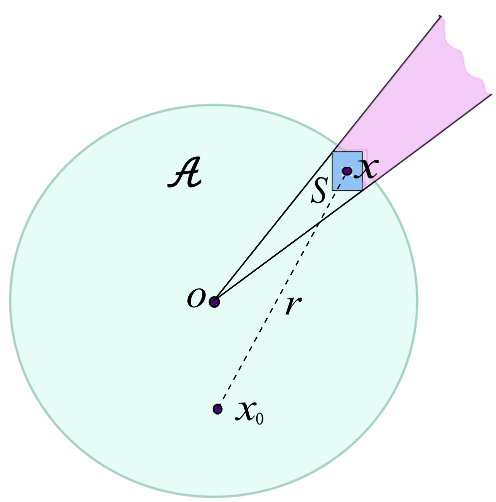



Fig. 17. Probability of $x$ in the truncated discrete laplacian.