Geo-Indistinguishability: Differential Privacy for Location-Based Systems

Miguel Andrés, Nicolás Bordenabe, Konstantinos Chatzikokolakis, Catuscia Palamidessi
INRIA, CNRS and LIX, Ecole Polytechnique

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

Our 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. We present three equivalent characterizations of this notion, one of which corresponds to a generalized version of the well-known concept of *differential privacy*. Furthermore, we present a perturbation technique for achieving geo-indistinguishability by adding (controlled) random noise to the user's location, drawn from a planar Laplace distribution. We demonstrate the applicability of our technique through two case studies: First, we show how to enhance applications for location-based services with privacy guarantees by implementing our technique on the client side of the application. Second, we show how to apply our technique to sanitize location-based sensible information collected by the US Census Bureau.

I. Introduction

The growing use of mobile devices equipped with GPS chips has significantly increased the use of location-based systems. This kind of systems employs geographical information (typically expressed as latitude and longitude coordinates) identifying the position of an entity in order to provide a service. Examples of location-based systems include (1) LBSs (Location-Based Services) such as mapping applications, GPS navigation, and location-aware social networks, as well as (2) location-data mining algorithms used to determine, among others, points of interest, traffic patterns, and disease geographical distributions.

While location-based systems have demonstrated to provide enormous benefits to individuals and society, their popularity raises important privacy issues. For example, by using an LBS, users may unknowingly allow companies to compile detailed profiles of their daily activities including places they visit, people they meet, and events they attend. Similarly, location-data mining algorithms can be used, for example, to determine the home location of individuals using their GPS navigation information.

In this work we present a novel technique that allows one to use LBSs while providing formal privacy guarantees to the users confiding the underlying geographical information. As a motivating example, consider a tourist in Paris who wishes to get information about restaurants nearby his current location, say, the Eiffel tower. Our mechanism works by first adding controlled noise to the user's location in order to obtain an approximate version of it, and then sending only the approximate location to the LBS. The kind of privacy that we aim at providing is *quasi-indistinguishability within a certain area*, which we will refer to, more briefly, as *geo-indistinguishability*. Intuitively, what it means is that, from the point of view of the attacher, the user could be anywhere – or more precisely, *almost equally likely to be anywhere* – within a certain radius r from the Eiffel tower. The value of r should be specified by the user, as well as the level of discrepancy ℓ that he can tolerate between the likelihood of the various points in the area (which are candidate locations for the adversary). These two values represent the privacy guarantees for the user. Clearly there is a trade-off between the desired level of privacy and the usefulness of the service provided by the LBS. For instance if the user wants to obtain only the restaurants in walking distance from the Eiffel tower, he should set r to be no more than, say, 1 kilometer. Hence these parameters should not be chosen arbitrarily. Also, they are related, in the sense that, the larger is the required area of protection, the more discrepancy we could tolerate between the likelihood of the various points, in order to maintain the same degree of usefulness, while still protecting the exact location. In general, with our method, when we specify a pair (ℓ, r) we obtain a mechanism that provides geo-indistinguishability for all pairs (ℓ', r') such that $\ell'/r' = \ell/r$.

We show that the notion of geo-indistinguishability can also be interpreted as the requirement that, within the radius r, the approximate location communicated by the user should not give too much hint to the adversary about his real location, where the "too much" is quantified by ℓ. Finally, we show that geo-indistinguishability can be seen as a generalization of the popular notion of *differential privacy*. This latter characterization emphasizes the fact that – like differential privacy – our notion is independent from the side information of the user, such as any prior probabilistic knowledge about the user's actual location.

Coming back to our mechanism for generating the approximate location, the inspiration comes from one of the main approaches used in differential privacy, which consists in drawing the noise from a Laplace distribution. This distribution, however, is linear, while we need a planar mechanism. Now, the Laplace distribution can be extended naturally on the continuous plane and it is easy to prove that such an extension provides the privacy guarantees we require. Thanks to a transformation to polar coordinates, we are also able to devise a simple and efficient method to draw points. However, as the common applications usually involve a finite representation of the coordinates, we need to discretize the distribution, and this operation raises serious concerns regarding the possible breach of geo-indistinguishability. Nevertheless, we prove that this property is still preserved, modulo a minor degradation of the level of privacy.

We conclude our work by demonstrating the applicability of our approach through two case studies, one based on LBS and the other on location-data mining. In the former case, we show that, by trading privacy for bandwidth usage, geo-indistinguishability can be obtained without degrading the utility of the information provided by LBS. In the latter case, we show how to apply our technique to sanitize datasets containing geographical information. In particular, we show how to sanitize publicly available geographic information released by the US Census Bureau. Our experiments reveal that providing geo-undistinguishability to all users in the dataset (i.e., US inhabitants) does not significantly decrease the quality of the sanitized data (the degree of decrease being inversely proportional to the parameters ℓ and r of the privacy guarantee).

In [1] a distinction is made between real-time systems (such as a location based service from a smartphone, which has to happen in real time) vs offline systems (like publishing statistical information about many users). Our two case studies show that our approach is suitable for both kinds of systems. However, it is in the domain of real-time systems that our notion of geo-indistinguishability and our mechanism shine: the notion because standard differential privacy is not applicable, and the mechanism because of its simplicity to implement in a smartphone. Because of this advantages we use the Paris-example as our motivating scenario trough the paper.

Road Map: In Section 2 we discuss privacy guarantees from the literature that could be potentially applied to Geolocation Systems and point out their weaknesses and strengths. In Section 3 we introduce the notion of geo-indistinguishability and highlight its relation with differential privacy. We then proceed to describe a mechanism that provides geo-indistinguishability in Section 4. In Sections 5 and 6 we demonstrate the applicability of our approach by case studies related to LBSs and Location-Data Mining, respectively. Section 7 discusses related work and Section 8 presents future work and conclusions.

II. Existing Notions of Location Privacy

In this section, we examine various notions of privacy from the literature, as well as techniques to achieve them. We use the example of a user accessing a location based service, already discussed in the introduction, as our motivating scenario. The user is visiting Paris, currently located near the Eiffel Tower, and wishes to find nearby restaurants with good reviews. To achieve this goal, he uses a handheld device (eg. a smartphone) to query a public LBS provider (such as Google Maps). However, the user expects his location to be kept private: informally speaking, the information sent to the provider should not allow him to accurately infer the user's location. Our goal is to provide a *formal* notion of privacy that adequately captures the user's expected privacy.

From the point of view of the mechanism for achieving privacy, we require a technique that can be performed in real-time by a handheld device such as a smartphone. We also require that no trusted anonymization party is involved, and optimally that no peer-to-peer communication with other users is needed. Such communication could be challenging if no other users are in proximity, and in most cases would require some level of trust between users which is challenging to achieve.

A. 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 ([3], [4], [5]) 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* ([6], [7]). 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 *cloacking* ([8], [9], [10]). This involves creating a cloacking region that includes k points sharing some property of interest, and then querying the service provider for this cloacking region.

The main drawback of k-anonymity-based approaches in general is that a system cannot be proved to satisfy this notion unless assumptions are made about the attacker's auxiliary 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 auxiliary 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, [6] takes into account concepts such as ubiquity, congestion and uniformity for generating dummy points, in an effort to make them look realistic. Similarly, [10] takes into account the user's auxiliary information to construct a cloacking region. Such counter-measures have their own drawbacks: first, they complicate the employed techniques, also requiring additional data to be taken into account, making their application in real-time by a handheld device challenging. Moreover, the attacker's actual auxiliary information might simply be inconsistent with the assumptions being made.

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

B. *Differential Privacy*

Differential Privacy ([11]) is a notion of privacy from the area of statistical databases. Its goal is to protect an individual's data while publishing aggregate information about the database. Differential privacy requires that modifying a single user's data should have a negligible effect on the query outcome. More precisely, it requires that the probability that a query returns a value v when applied to a database D, compared to the probability to report the same value when applied to an *adjacent* database D' – meaning that D, D' differ in the value of a single individual – should be within a bound of e^ϵ. A typical way to achieve this notion is to add controlled random noise to the query output, for example drawn from a Laplace distribution. An advantage of this notion is that a mechanism can be shown to be differentially private independently from any auxiliary information that the attacker might possess.

Differential privacy has also been used in the context of location privacy. In [12], it is shown that a synthetic data generation technique can be used to publish statistical information about commuting patterns, while satisfying differential privacy. In [13], a quadtree spatial decomposition technique is used to ensure differential privacy in a database with location pattern mining capabilities.

As shown by the aforemetioned works, differential privacy can be succesfully 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 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.

C. *Tranformation-based approaches*

A number of approches 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 tranforming all data to a different space, usually employing cryptographic techniques, so that they can be mapped back to spatial information only by the user ([14], [15]). 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 popular providers, such as Google Maps, which have access to the real data.

III. Geo-Indistinguishability

In this section, we define *geo-indistinguishability*, a formal definition of privacy which expresses a user's expected privacy requirement when using a location-based system. Our motivating scenario, already discussed in the previous sections, involves a user visiting Paris, currently located at a point x close to the Eiffel Tower, and wishing to discover nearby restaurants with good reviews. To achieve this goal, the user can query a service provider (for instance, an LBS server); however, to maintain his privacy, the user does not wish to disclose his exact location. Instead, he can provide some approximate information that still allows him to obtain a useful service, for instance, a randomly chosen point z close to his location. The question then is, what kind of privacy does the user *expect* to have in this scenario? On the one hand, the user clearly does not wish to reveal x. On the other hand, some approximate information is expected to be revealed. For instance, the fact that the user is located somewhere in Paris will be known by service provider, without being considered a violation of privacy (in fact this is desirable in order to obtain a useful service). But how can we formalize such a notion of privacy?

It is clear from the example above that privacy in this scenario depends on the accuracy with which the attacker can detect the user's location. To capture this property, the notion of privacy *within a radius* is crucial. We fix a circle of radius r centered at the user's location, and we reason about the user's level of privacy within this radius. Roughly speaking, we say that the user enjoys *ℓ-privacy within* r if, by observing z, the attacker's ability to detect the user's location *among all points within the radius* r, does not increase (compared to the case when z is unknown) by more than a factor depending on ℓ. The idea is that ℓ is the (inverse of) user's *level* of privacy for that radius: the smaller ℓ is, the stronger privacy the user enjoys (as it gets harder for the attacker to detect the user's location among the points within this circle). For the moment we keep this notion informal, it will be made precise later in this section.

Going back to the privacy requirements, the user does not wish to reveal his location x with high accuracy. Thus, the first important requirement is that within a short radius r_1 from x, for example 1 km, the user should enjoy ℓ_1-privacy for some small ℓ_1. This means that the user is well-protected within this area: knowing z does not allow the attacker to infer the user's location among the points within r_1.

But what about points outside r_1? Clearly, some privacy is still expected: a system which allows the attacker to infer with certainty that the user is located within r_1 is usually undesirable. Still, in order to allow the service provider to obtain approximate information about x necessary to provide its service, it is necessary to lower the privacy requirement outside this radius. Thus, we can select a larger radius r_2, say 5 km, and require the user to enjoy ℓ_2-privacy within r_2, for some slightly greater ℓ_2 (note that we simultaneously require privacy for both r_1 and r_2, only with different levels). Continuing this reasoning we can define gradually larger areas and drop the level of privacy each time, eventually becoming low enough so that the provider can infer that with high probability the user is somewhere in Paris (instead of, say, New York), but not its exact location.

So we see that the user's expected level of privacy is distance-dependent. Close to the user's location we require strong privacy, while more distant points are allowed to be distinguished. Going one step further in this reasoning, we can require that privacy holds for *any radius* r, with a level ℓ that is *proportional* to the radius, which brings to our first, still informal, definition of *geo-indistinguishability*:

> A mechanism satisfies ϵ-geo-indistinguishability iff for any radius $r > 0$, the user enjoys ϵr-privacy within r.

The parameter ϵ can be thought as the level of privacy at one unit of distance. This definition requires that the user is protected within any radius r, but with a level $\ell(r) = \epsilon r$ that increases with the distance. Within a small radius, for instance $r = 1$ km, $\ell(r)$ is small, guaranteeing that the attacker cannot infer the user's location within the 7th arrondissement of Paris. On the other hand, privacy decreases when we move away from the user's location; taking for instance $r = 10.000$ km, $\ell(r)$ becomes very large, allowing the provider to infer that with high probability the user is located in Paris instead of London.

Note that, for the user, a simple way to specify his privacy requirements is by a tuple (ℓ, r), where r is the radius he is mostly concerned with (eg. 1 km in the example above), and ℓ is the privacy level he wishes *for that radius*. In this case, it is sufficient to require ϵ-geo-indistinguishability for $\epsilon = \ell/r$; this will ensure a level of privacy ℓ 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 ℓ-privacy within r means. In the remaining of this section we formalize this notion in three different ways; all of them turn out to be equivalent, but they are all useful for understanding in depth the privacy guarantees provided by geo-indistinguishability.

A. Probabilistic model

We introduce here the simple probabilistic model that is 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, although for our needs we consider $\mathcal{Z}$ to also contain spatial points. In our operational scenario, the user is assumed to be located at some point $x \in \mathcal{X}$. He then selects a point $z \in \mathcal{Z}$ which is made available to the attacker (for instance, it is reported to an untrusted service provider).

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. Let X be the random variable giving the user's location (ranging over $\mathcal{X}$); the attacker's auxiliary information can be modelled by a *prior* distribution P_X for X, where $P_X(x)$ is the probability assigned to the location x.

Second, the selection of a point 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 IV). The probabilistic function for selecting a reported value based on the actual location is called a *mechanism*. Let Z be the random variable giving the reported point; a mechanism $\mathcal{K}$ for selecting z is a function assigning to each location $x \in \mathcal{X}$ a probability distribution for Z, where $\mathcal{K}(x)(S)$ is the probability that the reported point belongs to the set $S \subseteq \mathcal{Z}$, when the actual location is x.[1]

Together, P_X and $\mathcal{K}$ induce a *joint* probability distribution P for X, Z, as $P(x, S) = P_X(x)\mathcal{K}(x)(S)$. Note that, by construction, $P(x) = P_X(x)$ and $P(S|x) = \mathcal{K}(x)(S)$. We use the joint distribution P in all our definitions; still, we want the definitions to be independent from the attacker's side information. Thus, we either explicitely quantify over all priors P_X, or we use probabilities of the form $P(S|x)$ which depend only on the mechanism and not on the prior.

B. First approach

We return to the issue of formalizing what ℓ-privacy within a radius r means. An intuitive way of doing so, is to compare the probabilities of different points within r, after seeing a reported point in $S \subseteq \mathcal{Z}$ (note that we always consider sets of reported points, to allow for continuous

[1] For simplicity we assume X to be discrete, but allow Z to be continuous since we use continuous distributions in Section IV. Thus we need to talk about the probability of sets of points, implicitly assuming all mentioned sets to be measurable.

distributions). Let $x, x' \in \mathcal{X}$, such that $d(x, x') \leq r$, where $d(\cdot, \cdot)$ denotes the Euclidean distance between points. Ideally, we would like to require that ${P(x|S)}/{P(x'|S)} \leq e^{\ell}$, meaning that for a small ℓ, the attacker assigns similar probabilities to the user being located in x or x' after observing S.

However, we would like our definition to hold for any auxiliary information that the attacker might have, meaning for all priors P_X. Intuitively, we cannot expect the above condition to hold for all priors, since a point x (eg. the Eiffel Tower) with higher probability than x' (eg. a point in the Seine) will still have higher probability after observing S. In other words, if ${P(x)}/{P(x')}$ is large, we cannot expect the corresponding fraction after observing S to be small. What we can expect, however, is that the two fractions, before and after the observation, are similar, meaning that S has limited effect to the probabilities assigned by the attacker. This brings us to our first formal definition of geo-indistinguishability:

Geo-indistinguishability-I: A mechanism satisfies ϵ-geo-indistinguishability iff for all priors P_X and all observations $S \subseteq \mathcal{Z}$:[2]

$$\frac{P(x|S)}{P(x'|S)} \leq e^{\epsilon r} \frac{P(x)}{P(x')} \qquad \forall r > 0 \; \forall x, x' : d(x, x') \leq r$$

C. Second approach

A second approach for defining privacy within a radius r, is to focus on a single point and compare the probability of this point before and after the observation. Ideally, we would like to require that ${P(x|S)}/{P(x)} \leq e^{\ell}$, meaning that for a small ℓ, the probability of x should not be affected by the observation S. However, this requirement is clearly too strong since some information is allowed to be leaked: a point in Paris might have negligible prior probability, since the user could be located anywhere in the world, while after the observation its probability is substantially increased.

Remember, however, that we are interested in privacy *within the radius* r. Let $B_r(x)$ be the set of points at distance at most r from x. Since we are interested in the attacker's capability of locating the user withing this radius, we condition all probabilities on the event $B_r(x)$. In other words, we reason about how accurately the attacker could detect a particular point x, if he already knew that the point was within $B_r(x)$. This brings us to our second definition of geo-indistinguishability:

Geo-indistinguishability-II: A mechanism satisfies ϵ-geo-indistinguishability iff for all priors P_X and all observations $S \subseteq \mathcal{Z}$:

$$\frac{P(x|S, B_r(x))}{P(x|B_r(x))} \leq e^{\epsilon r} \qquad \forall r > 0 \; \forall x \in \mathcal{X}$$

D. Third approach

So far, we have considered the probability that the attacker assigns to points before and after observing S, since comparing these probabilities is a natural way to quantify how much S helps the attacker. We now change our standpoint and consider instead the probabilities of observations, instead of points. Intuitively, if two locations x, x' produce a reported value in S with similar probabilities, then S reveals little information about whether the actual location is x or x'. Thus, it is natural to require that ${P(S|x)}/{P(S|x')} \leq e^{\ell}$ for points that lie within the radius r. This brings us to our final definition of geo-indistinguishability:

Geo-indistinguishability-III: A mechanism satisfies ϵ-geo-indistinguishability iff for all observations $S \subseteq \mathcal{Z}$:

$$\frac{P(S|x)}{P(S|x')} \leq e^{\epsilon r} \qquad \forall r > 0 \; \forall x, x' : d(x, x') \leq r$$

This definition requires that points within distance 1 produce observations with similar probabilities. The farther away two points are, the more different we allow the probabilities of producing S to be. This is very similar to the definition of differential privacy, which requires two databases that differ on a value of a single user to produce the same answer with similar probabilities. Differential privacy aims at completely protecting the value of the user, since the value has limited effect on the probabilities of observations. In our scenario, however, such a requirement would be too strong. Since the only information is the location of a single user, differential privacy would require all locations to produce reported points with similar probability, thus it would be impossible for the service provider to extract any information about the user location. Nevertheless, in our case we do not want to completely hide the user's location, since some approximate information needs to be revealed in order to obtain the required service. Thus, the definition requires a level of privacy that depends on the distance between points.

Still, the connection between geo-indistinguishability and differential privacy is strong. In fact, the above definition can be rewritten as:

$$P(S|x) \leq e^{\epsilon d(x,x')} P(S|x') \qquad \forall x, x' \in \mathcal{X} \; \forall S \subseteq \mathcal{Z}$$

This is an instance of a generalized definition of differential privacy ([16]), taking into account an arbitrary metric between databases, where standard differential privacy corresponds to the so-called Hamming distance. Thus, ϵ-geo-indistinguishability can be though as differential privacy under the Euclidean metric.

[2] Note that for the sake of readability, we express the definitions in terms of fractions. To avoid issues with zero probabilities, we can write all definitions in flat form, i.e. $P(x|S)P(x') \leq e^{\epsilon r} P(x'|S)P(x)$.

Note that, although the generalized definition appears in the literature ([16], [17], [18]), it is usually treated as an intermediate step for achieving standard differential privacy, and little work has been done using metrics other than the Hamming distance for the privacy definition itself (the closest work in this direction being [18]). In particular, to our knowledge this is the first work considering differential privacy under the Euclidean metric, which is a natural choice for spatial data.

Finally, we can show that the three definitions of geo-indistinguishability given in this section are simply different ways of expressing the same privacy requirement.

Theorem 3.1: Geo-indistinguishability-I, II, III coincide.

A note on the unit of measurement: Since the notion of distance between points is crucial for the definition of geo-indistinguishability, a natural question is: how is the definition affected by the unit in which distance is measured? If d is the Euclidean metric expressed in meters, switching to some other unit means replacing d by $d' = k \cdot d$, in which all distances are scaled by a factor k (eg. $k = 1/1000$ for kilometers). However, the privacy guarantees of a mechanism are clearly not affected by such a change.

The point is that, since r is a physical quantity expressed in some unit of measurement, ϵ needs to be expressed in the inverse unit, so that $\ell = \epsilon\, r$ is a pure number. Thus, ϵ needs to be updated when changing the unit of measurement. For simplicity in the rest of this paper we will omit the unit of measurement when it is not important for the context, and we assume that, if the unit for r is some u, then the one for ϵ is $1/u$.

E. Protecting multiple locations

So far, we have assumed that the user has a single point 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, for instance the set of points itself, their centroid, etc. As in the case of a single point, privacy is still a requirement; the provider is allowed to obtain only approximate information about the points, their exact value should be kept private. In this section, we discuss how ϵ-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 of this section can be then directly applied to the case of multiple points, by using d_∞ as the underlying metric. Enjoying ℓ-privacy within a radius r means that the observation can help the attacker infer $\mathbf{x}$ among all tuples at distance r (i.e. tuples having all points at distance r from the corresponding points of $\mathbf{x}$), by a factor of at most e^l. All three definitions of geo-indistinguishability remain the same, the only change being the set of secrets and the distance between them.

Extending a mechanism to multiple points: A natural question then to ask is whether we can create a mechanism for tuples of points, by independently applying an existing mechanism to each individual point, and report a tuple of values. Let $\mathcal{K}_i, 1 \leq i \leq n$ be mechanisms for individual points. Starting from a tuple $\mathbf{x} = (x_1, \ldots, x_n)$, we independently apply $\mathcal{K}_i$ to x_i obtaining a reported points z_i, and then report the tuple $\mathbf{z} = (z_1, \ldots, z_n)$. Thus, the probability that the combined mehcanism $\mathcal{K}$ 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. $\mathcal{K}(\mathbf{x})(\mathbf{z}) = \prod_i \mathcal{K}_i(x_i)(z_i)$.[3]

The next question is what level of privacy does $\mathcal{K}$ satisfy. For simplicity, consider a tuple of only two points (x_1, x_2), to which the same mechanism $\mathcal{K}_0$, satisfying ϵ-geo-indistinguishability, is applied. At first look, one might expect the combined mechanism $\mathcal{K}$ to also satisfy ϵ-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 satisfy geo-indistinguishability for the same ϵ. On the other hand, $\mathcal{K}$ can be shown to satisfy a reduced level of privacy:

Theorem 3.2: If $\mathcal{K}_i$ satisfies ϵ_i-geo-indistinguishability for $1 \leq i \leq n$, then the combined mechanism $\mathcal{K}$ satisfies ϵ-geo-indistinguishability for $\epsilon = \sum_i \epsilon_i$.

Note that this issue is similar to the problem of composing queries in standard differential privacy. When the outcome of multiple queries is randomized by adding independent noise to each answer, the resulting mechanism satisfies differential privacy with a parameter ϵ which is the sum of the parameters of the individual mechanisms. The reason is exactly that the asnwers are correlated, since they come from the same database.

The technique of independently applying a mechanism to each point is useful when the number of points remains small, since privacy decreases with the number of points (a mechanism applied to n points satisfies $n\epsilon$-geo-indistingui-

[3] For simplicity we consider probabilities of points here; a formal treatment of continuous mechanism would require to consider sets.

shability). Still, this is sufficient for some applications, such as the case study of Section V. Note also that this technique is by no means optimal: similarly to standard differential privacy ([19], [20]), better results can be achieved by adding noise to the whole tuple $\mathbf{x}$, instead of each individual points. Developing such techniques for geo-indistinguishability is left as future work.

The case of uncorrelated points: In the previous paragraph we saw that independently applying a mechanism to multiple points can potentially decrease the level of privacy, due to the fact that the points can be correlated. On the other hand, we are sometimes interested in applying a mechanism to uncorrelated points, that is points that are either selected independently from each other, or for which we can assume that the attacker has no information about their correlation. This can be captured by requiring that the probability to select x_i is independent from x_j and vice versa, that is $P(\mathbf{x}) = \prod_i P(x_i)$ (note that $P(x_i)$ is still arbitrary). Under this restriction, an observation about x_j does not intuitively reveal any information about x_i. Assuming that $\mathcal{K}_i$ satisfies ϵ_i-geo-indistinguishability, it can be shown that the combined mechanism $\mathcal{K}$ satisfies the same level of privacy wrt the *individual point* x_i, that is $P(\mathbf{z}|x_i) \leq e^{\epsilon_i r} P(\mathbf{z}|x'_i)$ for all x_i, x'_i such that $d(x_i, x'_i) \leq r$. Note that the ϵ_i-geo-indistinguishability *might not* be satisfied for the tuple $\mathbf{x}$ (instead, we need to take $\epsilon = \sum_i \epsilon_i$ for the tuple, as shown in the previous section). Stil, assuming the lack of correlation, ϵ_i-geo-indistinguishability will be satisfied for each individual point x_i.

F. Comparison with standard differential privacy

As discussed in Section III-D, geo-indistinguishability is an instance of a generalized version of differential privacy, using the Euclidean metric to measure the distance between secrets. Thus, it is natural to examine how this notion compares to the one of standard differential privacy. As discussed in Section II, an advantage of geo-indistinguishability is that it can be applied to scenarios involving a single user, for which differential privacy is poorly suited. The comparison becomes more interesting in the case where secrets are tuples of n points, each corresponding to a different user. Note that we try to keep the discussion at a high level, focussing mainly on the privacy guarantees of each notion, and abstracting from the exact application.

Consider two mechanisms, $\mathcal{K}_1$ satisfying ϵ_1-geo-indistinguishability and $\mathcal{K}_2$ satisfying ϵ_2-differential privacy. Note that simply comparing ϵ_1, ϵ_2 is meaningless, since they refer to different definitions. To do a fair comparison, let $\mathbf{x} = (x_1, x_2, \dots, x_n)$, $\mathbf{x}' = (x'_1, x_2, \dots, x_n)$, be two tuples differing only in the location of the first user (i.e. seen as databases, they are *adjacent*). We then consider the level of privacy that each mechanism provides *for those tuples*, which corresponds to how well the secret of the first user is protected. The privacy levels ℓ_1, ℓ_2 of $\mathcal{K}_1, \mathcal{K}_2$ respectively, for those tuples, is:

$$\ell_1 = \epsilon_1 d_\infty(\mathbf{x}, \mathbf{x}') = \epsilon_1 d(x_1, x'_1) \qquad\qquad \ell_2 = \epsilon_2$$

in the sense that, for both mechanisms, the ratio $\mathcal{K}_i(\mathbf{x})(S) / \mathcal{K}_i(\mathbf{x}')(S)$ is bounded by e^{ℓ_i} for all observations S. Thus, comparing the two mechanisms boils down to comparing ℓ_1, ℓ_2, for various points x_1, x'_1.

An important observation is that ℓ_2 is independent from the actual points x_1, x'_1. This means that standard differential privacy protects all values in the same way; any secret value of a user is equally indistinguishable from any other. This is not the case for ℓ_1, however, which depends on the actual points x_1, x'_1, and more precisely on their distance. So, the level of protection depends on the secrets; the closer two points are the harder it is for the attacker to distinguish them.

Thus, for points far away from each other, ℓ_1 will be greater than ℓ_2, so differential privacy offers better protection, while geo-indistinguishability becomes better in points close to each other, for which ℓ_1 is smaller than ℓ_2. This behaviour becomes more important in cases where ϵ_2 is not very small, which is often unavoidable in order to provide acceptable utility (see, for instance, Section VI). Intuitivelly, when ϵ_2 is large, then offering the same protection $\ell_2 = \epsilon_2$ for all points becomes a drawback. A privacy level that depends on the distance ensures that nearby points (which, in the case of location-based systems, need to be highly indistinguishable), will be adequately protected.

Finally, when comparing notions of privacy, 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.

A note on auxiliary information: An important advantage of differential privacy, compared to other privacy notions, is that it provides privacy guarantees independently from the attacker's auxiliary information. This does not mean that auxiliary information is useless to the attacker: consider for instance an "average height" query; information such as "all users have the same height" or "Alice is 5 cm taller than the average", together with the mechanism's output z (i.e. the randomized average), helps narrowing down the probable values of Alice's height. What differential privacy actually guarantees is that, under any auxiliary information (expressed in terms of a priori probability on Alice's value), learning z does not reveal much more information about this value (where "much more" is measured by ϵ).

Being an instance of differential privacy under the Euclidean metric, geo-indistinguishability has a similar behavior wrt auxiliary information. Namely, it guarantees (as shown by the formulation of Section III-C) that under any auxiliary information (expressed by the a priori on the actual location), learning z does not reveal much more information about x (where "much more" is measured by ϵr).

In the case of geo-indistinguishability, we have an additional privacy guarantee: since $P(z|x) \leq P(z|x, B_r(x))$, a geo-indistinguishable mechanism guarantees that learning z does not reveal much more information about the user's location x than learning that the user is within the circle $B_r(x)$. This property does not seem to have a counterpart in differential privacy.

IV. A MECHANISM FOR GEO-INDISTINGUISHABILITY

In this section we present a method to generate noise in a way that satisfies geo-indistinguishability. We model the location domain as the Euclidean plane equipped 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".

For applications with digital interface the domain of interest is discrete, since the representation of the coordinates of the points is necessarily finite. However, it does not seem easy to devise an efficient mechanism for geo-indistinguishability that generates noise directly on a discrete plane (we will come back to this point in Section IV-B). We therefore consider a different approach:

(a) First, we define a geo-indistinguishable, continuous mechanism for the ideal case of the continuous plane.

(b) Then, we discretized the mechanism by remapping each point generated according to (a) to the closest point in the discrete domain.

Furthemore, we may want to consider only a limited area. For instance if we are in a island, we may wish to report only locations in the land, not in the sea. Thus we may want to apply a third step:

(c) If desirable, we may truncate the mechanism, so to report only points within the limits of the area of interest.

A. A geo-indistinguishable continuous mechanism

In this section we explore how to define a geo-indistinguishable mechanism on the continuous plane. This will constitute the basis of our method.

The idea is that whenever the actual location is $x_0 \in \mathbb{R}^2$, we report, instead, a point $x \in \mathbb{R}^2$ generated randomly according to the noise function. The property that we need to guarantee is that the probabilities of reporting a point in a certain (infinitesimal) area around x when the actual locations are x_0 and x'_0 respectively, should differ at most by a multiplicative factor $e^{-\epsilon\, d(x_0, x'_0)}$.

Intuitively, this property is achieved if the noise function is such that the probability of generating a point in the area around x decreases exponentially with the distance from the actual location x_0. In a linear space this is exactly the behavior of the Laplace distribution, whose probability density function (pdf) is:

$$\frac{1}{2\,b}\, e^{-\frac{|x-\mu|}{b}} \tag{1}$$

where μ is the expected value, in this case set to be the actual location, and b is a parameter. This function has been used in the literature to define an oblivious mechanism for adding noise to queries on statistical databases, with μ set to be the actual answer, and it has been proved that such mechanism provides ${}^{1}\!/_{b}$-differential privacy [21]. Figure 1 illustrates the idea.

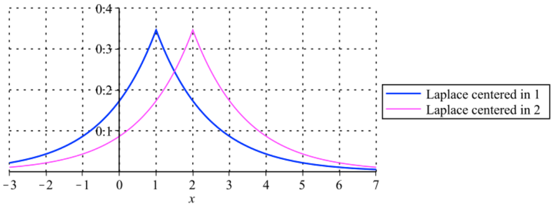

Figure 1. The distributions defined by two linear laplacians, centered in 1 and in 2 respectively, with $1/b = \ln 2$. The ratio between the two curves is at most $e^{\frac{1}{b}} = 2$ everywhere.

Of course we cannot use the standard Laplace distribution for our purposes, because it is defined on the line, while we need a distribution defined on the plane. Furthermore we need to use the (Euclidean) planar distance $d(x, \mu)$ instead of the liner distance $|x - \mu|$. Intuitively, however, just replacing $|x - \mu|$ by $d(x, \mu)$ in (1) results into a natural extension of the Laplace distribution from one to two dimensions[4]. We call *planar laplacian* such extension.

The probability density function: Given the parameter $\epsilon \in \mathbb{R}^+$, and the actual location $x_0 \in \mathbb{R}^2$, the pdf of our noise mechanism, on any other point $x \in \mathbb{R}^2$, is:

$$D_\epsilon(x_0)(x) = \frac{\epsilon^2}{2\,\pi}\, e^{-\epsilon\, d(x_0, x)} \tag{2}$$

where ${}^{\epsilon^2}\!/_{2\,\pi}$ is a normalization factor. Using a transformation in polar coordinates it is possible to show that the integral

[4] In the literature there are various proposals for the extension of the Laplace distribution to higher dimensions. These are called *multivariate laplacians*. In general *multivariate* means that it involves $k \geq 1$ random variables. The particular cases of $k = 1$ and $k = 2$ are called *univariate* and *bivariate* respectively. Our definition corresponds to a particular instance of the extension investigated in [22], [23]. The same instance has been adopted also in [24].

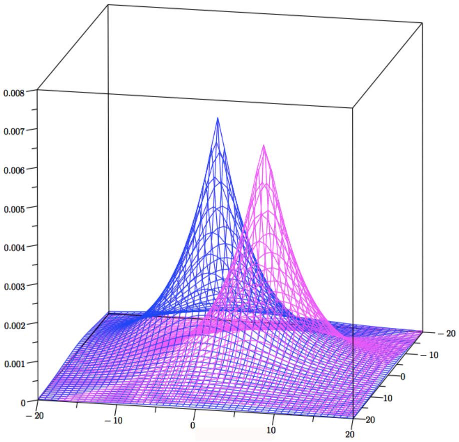

Figure 2. The pdf's of two planar laplacians, centered in $(-2,-4)$ and in $(5,3)$ respectively, with $\epsilon = 1/5$. The distance between the centers is $7\sqrt{2}$, and the ratio between the curves is at most $e^{7/5\,\sqrt{2}} \approx 7.24$ everywhere.

of this function over the whole $\mathbb{R}^2$ gives 1, which means that it is indeed the pdf of a probability distribution.

We call this function *planar laplacian centered in* x_0. The corresponding distribution is illustrated by Figure 2. Note that the projection of a planar laplacian on any vertical plane passing by the center gives a graph proportional to the one of a linear laplacian (Figure 1).

In Appendix B we show that the mechanism defined by a planar laplacian satisfies ϵ-geo-indistinguishability.

Drawing a random point: We illustrate now how to draw a random point from the pdf defined in (2).

First of all, we note that the pdf of the planar laplacian depends only on the distance from x_0. It will be convenient, therefore, to transform the reference system into a system of polar coordinates with origin in x_0. Intuitively, in this way the pdf will depend only on one variable, thus simplifying the drawing procedure.

So, given the pdf in (2), we consider the transformation into a system of polar coordinates (r,θ) where r is the radius and θ is the angle. A point x in cartesian coordinates will be represented as a point (r,θ) in the new system, where r is the distance of x from x_0, and θ is the angle that the line $x\,x_0$ forms with respect to the axis x of the cartesian system. Following the standard transformation method, the pdf of the *polar laplacian* centered in the origin (x_0) is:

$$D_\epsilon(r,\theta) = \frac{\epsilon^2}{2\,\pi}\, r\, e^{-\epsilon\, r} \qquad (3)$$

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 Θ (the angle). The two marginals are:

$$\begin{array}{rcl} D_{\epsilon,R}(r) & = & \int_0^{2\pi} D_\epsilon(r,\theta)\, d\theta \;=\; \epsilon^2\, r\, e^{-\epsilon\, r} \\ D_{\epsilon,\Theta}(\theta) & = & \int_0^{\infty} D_\epsilon(r,\theta)\, dr \;=\; \frac{1}{2\,\pi} \end{array}$$

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* $1/\epsilon$. Figure 3 shows the graph of this function for various values of ϵ.

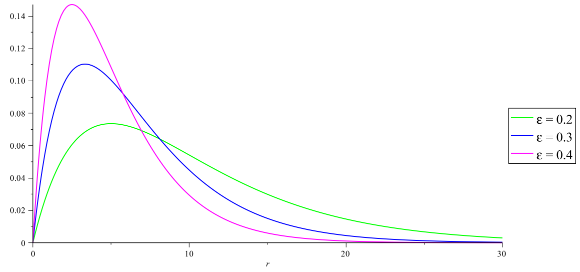

Figure 3. Pdf of the gamma distr. ($D_{\epsilon,R}(r)$) for various values of ϵ.

It may come as a surprise that this graph differs significantly from those in Figures 1 and 2, and in particular, that it does not have its maximum in the origin. Remember, however, that the graph in Figure 3 represents a pdf *in polar coordinates*. More precisely, $D_{\epsilon,R}(r)$ represents the probability that the random point is located in the circular crown centered in the origin and delimited by r and $r+dr$. The area of this crown is proportional to r, hence when r is close to 0 also the probability is close to 0. As r increases the probability increases, until the factor $e^{-\epsilon\, r}$ takes over. For r approaching infinity, the factor $e^{-\epsilon\, r}$ approaches 0, and dominates over r, hence the probability approaches 0 again.

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

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

We now show how to draw r. Following standard lines, we consider the cumulative function $C_\epsilon(r)$ of $D_{\epsilon,R}(r)$:

$$C_\epsilon(r) \;=\; \int_0^r \epsilon^2 \rho\, e^{-\epsilon\rho} d\rho \;=\; 1-(1+\epsilon\, r)\, e^{-\epsilon\, r}$$

Intuitively, $C_\epsilon(r)$ (see Figure 4) represents the probability that the radius of the random point falls between 0 and r. Finally, we generate a random number z with uniform probability in the interval $[0,1)$, and we set $r = C_\epsilon^{-1}(z)$.

Given a "universal" cartesian reference system and the actual location $x_0 = (s,t)$ in this system, if we could work in the "ideal" continuous plane, then we would just need to generate the noise (r,θ) as specified above, and then reports the point $x = (s + r\cos\theta, t + r\sin\theta)$. In practice however there is always some discretization involved, because (a)

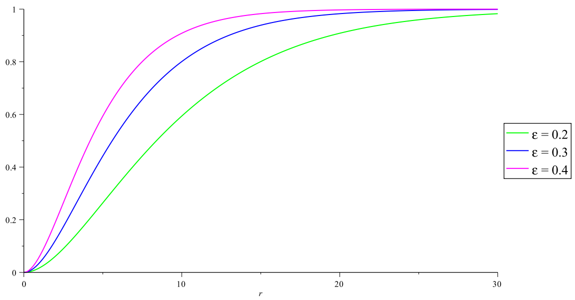

Figure 4. Plot of $C_\epsilon(r)$ for various values of ϵ.

Drawing a point (r, θ) from the polar laplacian

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

Figure 5. Method to generate the noise

computers have finite precision, and (b) (more important) the coordinates of the "universal reference system" will have a finite representation, typically using only a few decimal digits. The discretization of our method, and its properties, constitute the subject of next section.

B. Discretization

In real-life situations usually we represent a location by means of discrete coordinates. For instance, latitude and longitude up to some decimal of precision. Thus we study here how to define an approximation of the Laplace distribution on a grid $\mathcal{G}$ of discrete cartesian coordinates. Again, the property that we need to preserve is that the probability of generating a point x in the grid decreases exponentially with the distance from the actual location x_0.

Before we start illustrating our method, we wish to explain why we did not adopt the following approach, which seems the most natural: In the univariate case, the discrete approximation of the Laplace distribution is the double geometric probability distribution $\lambda e^{-\epsilon\,|x-x_0|}$, where $x \in N$ and λ is a normalization factor. This probability function can be visualized as a symmetric series of "steps" exponentially decreasing with the (discrete) distance from x_0. The obvious extension to the bivariate (discrete) case would then be the probability distribution $K(x_0)(x) = \lambda' e^{-\epsilon\, d(x_0,x)}$ where λ' is a suitable normalization factor.

Unfortunately, *there does not seem to be an efficient way to draw points according to the above distribution.* For this reason we propose a different approach, that can be summarized as follows. Given the actual location x_0, we report the point x in $\mathcal{G}$ obtained in the following way:

(a) first, we draw a point (r, θ) from the polar laplacian centered in x_0 (see (3)), as described in Figure 5,
(b) then, we remap (r, θ) to the closest point x on $\mathcal{G}$.

We will denote by $K : \mathcal{G} \to \mathcal{P}(\mathcal{G})$ the above mechanism. In summary, $K(x_0)(x)$ represents the probability of reporting the point x when the actual point is x_0.

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

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

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, θ) is drawn with the probability of the area between r, $r + \delta_r$, θ and $\theta + \delta_\theta$, where δ_r and δ_θ 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 scaly region associated to $R_{\mathcal{W}}(x)$ has a different shape and area depending on the position of x relatively to x_0.

Figure 6 illustrates the situation. The cartesian grid constituted by blue horizontal and vertical lines represents $\mathcal{G}$. The polar grid constituted by black circles and radial lines represent $\mathcal{W}$. The two dashed rectangles around the points x_0 and x_1 represent $R(x_0)$ and $R(x_1)$. The regions R_0 and R_1 colored in grey and magenta correspond to $R_{\mathcal{W}}(x_0)$ and $R_{\mathcal{W}}(x_1)$ respectively. Note that R_0 and R_1 have different shapes and areas, for instance R_0 is larger than R_1.

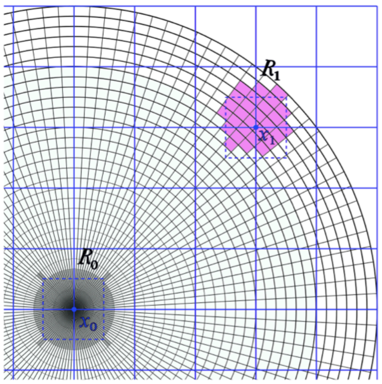

Figure 6. Remapping the points in polar coordinates to points in the grid

In the next paragraph we show that, despite of the above problem, the discretization *does* preserve geo-indistinguishability, under some moderate restrictions.

Geo-indistinguishability of the discretized mechanism: We now analyze the privacy guarantees provided by our discretized mechanism. We will show that the discretization essentially preserves geo-indistinguishability, although there is some degradation of the parameter ϵ. More precisely we will show that, after the discretization, our mechanism still satisfies ϵ'-geo-indistinguishability, within a range $r_{\max}$, for a suitable ϵ' that depends on ϵ, on the length of the step units of $\mathcal{G}$, and on the precision of the machine. 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 δ_θ and by δ_r denote the precision of the machine in representing θ and r, respectively.

The following theorem, whose proof is in Appendix C, states the property of geo-indistinguishability provided by our mechanism.

Theorem 4.1: Let $r_{\max} = {}^{u}\!/_{q\,\delta_\theta}$, where $q > 5$, and assume $\delta_r \leq {}^{u}\!/_{q}$. Given $\epsilon \in \mathbb{R}^+$, let ϵ' be defined as

$$\epsilon' = \epsilon + \frac{1}{u} \ln \frac{q - 2 + 3\, e^{\epsilon\, v \sqrt{2}}}{q - 5}$$

Then K provides ϵ'-geo-indistinguishability in any area of diameter $r_{\max}$, namely:

$$K(x_0)(x) \leq e^{\epsilon'\, d(x_0, x_0')} K(x_0')(x)$$

whenever $d(x_0, x), d(x_0', x) \leq r_{\max}$.

Figure 7 shows the value of ϵ' as a function of ϵ, for various grid units u, v and various values of q. (The graph does not depend on the unit of measurement of δ_r and u, but they have to be the same, and that of ϵ' and ϵ has to be the inverse, as usual.) As we can see, the degradation of the privacy level is not too serious, especially for large values of q (i.e. $q \geq 10^5$). Note that with double precision δ_θ is about $7\,10^{-16}$, hence choosing $q = 10^5$ means to set $r_{\max}$ to more than 10^{10} times u.

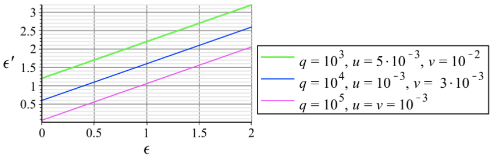

Figure 7. ϵ' as a function of ϵ for various precisions and grid units.

Note that in Theorem 4.1 the restriction to the area of diameter $r_{\max}$ is crucial. Namely, ϵ'-geo-indistinguishability does not hold for arbitrary distances for any finite ϵ'. Intuitively, this is because the step units of $\mathcal{W}$ (see Figure 6) 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 for 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.

On the other hand, the restriction about $r_{\max}$ is not a strong limitation, because the distribution decreases exponentially with r, and $r_{\max}$ is usually large, hence the points with distance $r > r_{\max}$ have anyway negligible probability. Also the assumption that $u \geq 5\,\delta_r$ is not restrictive for the kind of applications we are targeting, where u is several orders of magnitude larger than δ_r.

C. Truncation

In real-life applications usually we are interested in locations within a certain region. The laplacian mechanisms described in previous sections, however, feature the capability to generate points everywhere in the plane. If the user knows that the actual location is situated within a certain region, it seems desirable that the reported location lies within the same region as well, or at least not too far apart. To this purpose we propose a variant of the discrete laplacian described in previous section, which generates points only within a specified region.

We assume that the specified region $\mathcal{A}$ of acceptable report points is a circle centered in o, and diameter $diam(\mathcal{A})$. Our mechanism works like the discretized laplacian of previous section, with the difference that, whenever the point generated in step (a) lies outside $\mathcal{A}$, we remap it to the closest point in $\mathcal{A} \cap \mathcal{G}$ (which necessarily will be on the perimeter of $\mathcal{A}$, modulo discretization).

Let us denote by K^T the truncated variant of the mechanism K described in previous section. The type is: $K^T : \mathcal{A} \to \mathcal{P}(\mathcal{A} \cap \mathcal{G})$ and the drawing is described by the following procedure. Given the actual location $x_0 \in \mathcal{A}$:

(a) first, draw a point (r, θ) from the polar laplacian centered on x_0, as explained in previous section,

(b$'$) then, remap (r, θ) to the closest point x on $\mathcal{A} \cap \mathcal{G}$.

Intuitively, K^T behaves like K except when the region $R(x)$ is on the border of $\mathcal{A}$. In this case, the probability on x is given not only by the probability of the points in $R_{\mathcal{W}}(x)$, but also by the probability of the part of the cone determined by o and $R(x)$ which lies outside $\mathcal{A}$.

We are now going to show that this new method satisfies geo-indistinguishability on all $\mathcal{A}$, provided that $r_{\max}$ is not smaller than $diam(\mathcal{A})$. The proof is in Appendix D.

Theorem 4.2: If $r_{\max} \geq diam(\mathcal{A})$, then K^T provides ϵ'-geo-indistinguishability, namely

$$K^T(x_0)(x) \leq e^{\epsilon'\, d(x_0, x_0')} K^T(x_0')(x) \quad \text{for every } x_0, x_0' \in \mathcal{A}$$

where $r_{\max}$ and ϵ' are defined as in Theorem 4.1.

V. Enhancing Location-Based Services with Privacy

The growing use of mobile devices equipped with GPS chips in combination with the increasing availability of wireless and GSM connection has significantly increased the use of LBSs. A resent study in the US shows that 46% of the adult population of the country owns a smart-phone and, furthermore, that 74% of those owners use LBSs [25]. Examples of LBSs include mapping applications (eg, Google Maps and Bing Maps), Points of Interest (POI) retrieval (eg, AroundMe and Localscope), coupon/discount providers (eg, GroupOn and Yowza), GPS navigation (eg, TomTom and Google Maps), and Location-Aware social networks (eg, Foursquare and OkCupid).

Users invoking a LBS typically submit their location in order to obtain a certain benefit, eg, information about POI in the area around them. Although LBSs have proved to offer important benefits for a variety of applications, the privacy exposure of users' location information is undeniable and, unfortunately, often overlooked. LBS providers can collect accurate location information about users and, potentially, process it enabling them to infer sensitive information such as users' home location, work location, sexual preferences, political views, and religious inclinations.

In this section we show how to enhance LBS applications with privacy guarantees while still providing a high quality service to their users.

A. Geo-indistinguishability for POI retrieval LBSs

Let us start by describing how geo-indistinguishability can be used to specify a subtle notion of privacy for LBS applications. For that purpose, we first delineate the architecture of LBS applications that we consider in this work. 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, our approach does not rely on trusted third-party servers (in contrast with several solutions proposed in the literature, see Section II). Additionally, since this work focusses on the potential harm incurred to users by conferring their location to a LBS, we assume that users only communicate location information to the provider (although typically more information, such as user ID and network address, is transmitted). Figure 8 illustrates the LBS setting that we consider in this work.

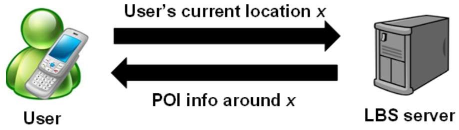

Figure 8. LBS architecture

For illustration purposes, in this section we will focuss on LBSs applications providing POI information. However, most of the discussion and techniques presented in the following, hold for a broader family of LBS applications (some of which we mention explicitly below).

Coming back to our running example, we now study how geo-indistinguishability can help to provide privacy guarantees to the user visiting Paris. More precisely, let us assume that the user is sitting at Café Les Deux Magots and wishes to obtain information about nearby restaurants without revealing to a potential attacker (the LBS provider in this case) his exact location. However, as discussed before, in order to obtain accurate information from the LBS provider, the user is willing to reveal some approximate information. This privacy guarantee can be captured by our notion of ϵ-geo indistinguishability. Letting the user specify his desired level of privacy, say $\ell = \ln(4)$ within $r = 0.2$ km (and decreasing proportionally for larger distances), $\ln(4)/0.2$-geo-indistinguishability guarantees the user that by using the LBS application (and thus revealing his approximate location), the LBS provider cannot infer his real location (at least not with probability 4 times higher than without revealing his location) among all locations within 200 meters.

B. Privately Retrieving POI information from a LBS

We now proceed to describe how to enhance LBS applications with geo-indistinguishability guarantees. 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 (forecast information for an approximate location is typically as good as forecast information for an exact location), location-aware advertising/offers (eg, shops offering discounts typically care about users being nearby – rather than their exact location), and a number of LBS applications for POI retrieval (eg, retrieving nearby cheap gas stations or nearby tourist sites when visiting a city). Enhancing this kind of LBSs with geo-indistinguishability privacy guarantees is relatively straightforward. It requires to implement the location perturbation mechanism presented in Section IV on the client party of the LBS application and then report the generated approximate location (instead of the real location) to the LBS server party. We note that this simple modification does not require a significant computation overlay on the client side nor extra bandwidth usage.

For highly-location-sensitive LBS applications, on the other hand, the quality of the service provided by LBSs highly depends on the precision of the location information submitted by the user. Our running example lies within this 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 this kind of LBS applications with privacy guarantees is considerably more challenging. In the following we describe how to enhance this kind of LBS applications with privacy guarantees while still providing a high quality service. Our approach requires three modifications to the standard LBS architecture:

1) The mechanism described in Section IV should be implemented on the client application in order to report to the LBS server party the user's approximate 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

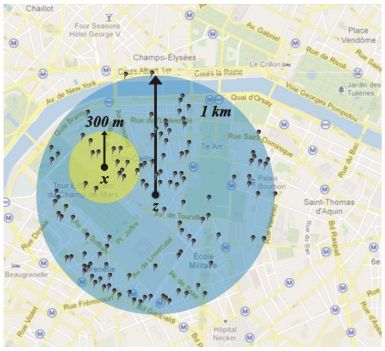

Figure 9. Retrieval information situation for private LBS

the user wishes to obtain information about POI within, say, 300 meters of x, the client application should request information about POI within, say, 1 km of z. This situation is depicted in Figure 9 for our running example. The user's current location x is at café Les Deux Magots and the reported approximate location z submitted by the client application is at about 600 meters from x. We will refer to the circle centered at x with 300 meters radius as *area of interest* (of the user) and to the circle centered at z with 1 km radius as *area of retrieval*.

3) Finally, the client application should filter the retrieved POI information (depicted by the pins within the area of retrieval in Figure 9) in order to provide to the user with the desired information (depicted by pins within the user's area of interest in Figure 9).

The resulting client-server communication is shown in Figure 10.

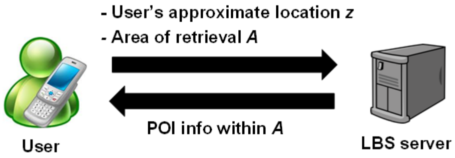

Figure 10. LBS architecture

Clearly, for our approach it is crucial that the area of interest is fully contained in the area of retrieval (otherwise the information expected by the user might not be fully retrieved from the server). However, the latter depends on a randomly generated location, hence such condition cannot be guaranteed (at least not with probability 1). Note that the client application could dynamically adjust the area of retrieval in order to ensure that it always contains the area of interest. However, this approach would jeopardize the privacy guarantees: on the one hand, the size of the area of retrieval 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. Therefore, in order to provide geo-indistinguishability in this setting, the area of retrieval should be defined *independently* from the randomly generated location.

Our approach consists on *statically* defining the area of retrieval as a function of the security parameters (ℓ and r) and of the area of interest. Our goal is to define an area of retrieval as small as possible (in order to avoid retrieving unnecessary information and, consequently, unnecessary bandwidth usage) in a way that the area of interest is contained in it with probability as high as possible. Since such goal highly depends on the *accuracy* of the mechanism generating the approximate location (ie, on how close the generated location and the real location are to each other) before presenting our solution we need to introduce the notion of accuracy for data sanitation mechanisms.

C. Accuracy for location perturbation mechanisms

As it is standard for privacy enhancing mechanisms based on data perturbation (eg, the Laplacian mechanism providing standard differential privacy [21]), the aim of our mechanism is to provide accurate (location) information in a private way (ie, while satisfying geo-indistingushability).

In order to evaluate the accuracy of our mechanism, we will use a well-known concept from the literature (adapted to our location setting) that aims at assessing the accuracy of the (approximate) information generated by a mechanism, (α, δ)-usefulness [19]:

A location perturbation mechanism $\mathcal{K}$ is (α, δ)-*useful* if for every location x, with probability at least $1 - \delta$, the reported location $z = \mathcal{K}(x)$ satisfies

$$d(x, z) \leq \alpha.$$

Therefore, a (α,δ)-useful mechanism generates approximate locations z within distance α of the exact location x with probability at least $1 - \delta$. Figure 11 illustrates how our mechanism behaves with respect to (α,δ)-usefulness when providing ϵ-geo-indistinguishability for $r = 0.2$ (as in our running example) and several values of ℓ.

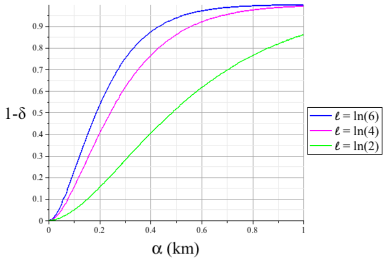

Figure 11. (α, δ)-usefulness for $r = 0.2$ and various values of ℓ.

It follows from the information in Figure 11, that a mechanism providing the privacy guarantees specified in our running example (ϵ-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 0.99, 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 define an area of retrieval containing the area of interest with a given probability. Note that an area of retrieval with radius, say, r_A contains the area of interest with radius say, r_I, with probability at least $1-\delta$ if the mechanism used to generate the reported location is (α,δ)-useful, for an $\alpha \leq r_A - r_I$.

Therefore, by setting r_A to 1 km in our running example and since our mechanism is (0.69, 0.05)-useful, it is guaranteed that the retrieval area contains the area of interest with probability at least 0.95.

D. Further challenges: using a LBS multiple times

After describing how to provide geo-indistinguishability guarantees to users querying a LBS application a *single* time, we now discuss how to extend our solution to the case in which users wish to perform *multiple* queries.

In this scenario, the mechanism should protect multiple locations rather than one. But, what does it mean to enjoy privacy for multiple locations? As discussed in Section III-E, geo-indistinguishability can be naturally extended to this scenario. In short, the idea of being *ℓ-private within* r remains the same but for all locations simultaneously. In this way the locations, say, x_1, x_2 of a user employing the LBS twice remain indistinguishable from all pair of locations at (point-wise) distance at most r (ie, from all pairs x_1', x_2' such that $d(x_1, x_1') \leq r$ and $d(x_2, x_2') \leq r$).

A simple way of obtaining geo-indistinguishaility guarantees when performing multiple queries is to employ the our technique for protecting single locations to *independently* generate approximate locations for each of the user's locations. In this way, a user performing n queries via a mechanism providing ϵ-geo-indistinguishability enjoys $n\epsilon$-geo-indistinguishability (see Theorem 3.2).

This solution might result satisfactory when the number of queries to perform remains fairly low, but, due to the privacy degradation, impractical otherwise. It is worth noting that the canonical technique for achieving standard differential privacy (based on adding noise according to the Laplace distribution) suffers of the same privacy degradation problem (query composition degrades privacy linearly on the number of queries). Several articles in the literature focus on this problem (see [20] for instance). We believe that the principles and techniques used to deal with this problem for standard differential privacy could be adapted to our scenario (either directly or motivationally). A fruitful direction to explore, in our particular scenario, is to employ the location history of the user together with the corresponding locations reported to the LBS provider (ie, (x, z) pairs) to "adjust" the way approximate locations are generated (eg, report z whenever the user's location x' is nearby a location x that the mechanism has previously reported as z). This challenge constitutes our main focus for future work.

VI. Sanitizing datasets: US census case study

In this section we present a sanitation algorithm for datasets containing geographical information. Roughly speaking, the algorithm iteratively sanitizes each of the geographic sensitive values in the dataset by means of the perturbation technique presented in Section IV.

A. The LODES dataset

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.

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.

Due to privacy constraints and legal issues, data involving the residence location of individuals cannot be released without previous sanitation; thus, the LODES dataset is a sanitized version of the real data. However, for illustration purposes and wlog, in the remaining of this section we will treat the LODES dataset as if it were the real data. The Census Bureau uses a *synthetic data generation algorithm* [26], [12] 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. For that purpose, a statistical model is built from the LODES dataset and then a sanitized counterpart is obtained by sampling points from the model.

In the following we present a sanitizing algorithm for datasets with geographical information (eg, the LODES dataset) that provides formal privacy guarantees. In particular, our algorithm provides geo-indistinguishability guarantees under the assumption that the home census blocks values in the dataset are uncorrelated (see the discussion about uncorrelated points in Section III-E). Although this assumption weakens the privacy guarantees provided by geo-indistringuishability, 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 ($hCoord$, $work\_coord$) pairs within small areas contained in the dataset, a practical attack based on correlation of points results highly improbable.

B. The Sanitizing Algorithm

The algorithm, illustrated in Figure 12, takes as input (1) a dataset D to sanitize, (2) the privacy parameters ℓ and r (see Section III), and (3) the precision parameters u, δ_r and δ_θ (see Section IV-B). The output of the algorithm is the sanitized counterpart of D. The algorithm is guaranteed to provide $\frac{\ell}{r}$-geo-indistinguishability to the home coordinates of all individuals in the dataset (see discussion on protecting multiple locations in Section III-E).

In order to sanitize D, our algorithm first determines the level of protection ϵ' needed in order to guarantee $\epsilon = \ell/r$ protection (see Theorem 4.1), and the acceptable region $\mathcal{A}$ (see Section IV-C). We assume that $\mathcal{A}$ contains all the home locations in D and has a diameter not larger than $r_{\max} = u/\delta_\theta$. Then for each $(c_h, c_w) \in D$, a radius ρ and an angle θ are drawn following the method in Figure 5. The noise (ρ, θ) is then used to generate a sanitized home coordinate c'_h on the reference system (see Section IV-B). Next, a test verification is performed in order to guarantee that c'_h lies in $\mathcal{A}$ and, in case it does not, c'_h is remapped to the closest point in $\mathcal{A}$ (see Section IV-C). Finally, the new sanitized pair (c'_h, c_w) is added to the output dataset D'.

```
Sanitizing Algorithm
Input: $D : hCoord \times wCoord$ // dataset to sanitize
       $\ell$, $r$ // privacy parameters
       $u$, $\delta_r$, $\delta_\theta$ // precision parameters
Output: Sanitized version $D'$ of input $D$
  $D' = \emptyset$; // initializing output dataset
  $\epsilon' = safe\_\epsilon(\ell, r, u, \delta_r)$; // Theorem 4.1
  $\mathcal{A} = acceptRegion(u, \delta_\theta)$; // acceptable report points
  for each $(c_h, c_w) \in D$ do
      Draw radius $\rho \sim gamma(2, 1/\epsilon')$;
      Draw angle $\theta \sim Uniform(2\pi)$;
      $c'_h = destinationPt(c_h, \rho, \theta)$; // sanitized point
      if $c'_h \notin \mathcal{A}$ then $c'_h = closestPt(\mathcal{A}, c_h, \rho, \theta)$; // truncate
      $D' = D' \cup \{(c'_h, c_w)\}$; // adding sanitized point
  end-for
  return $D'$;
```

Figure 12. Our sanitizing algorithm, based on data perturbation

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.

C. Experiments

In order to evaluate the accuracy of the sanitized dataset generated by our algorithm (and thus of our algorithm as a data sanitizer) we have 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.

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 experiments 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 sanitised versions of it generated by our algorithm. Figure 13 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 ℓ. 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 δ_r and δ_θ (for the corresponding ranges).

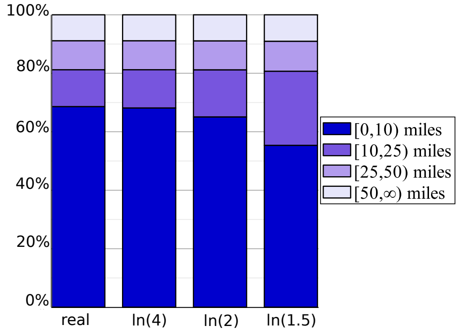

Figure 13. Home-work commute distance for $r = 1.22$ and various ℓ.

We have also conducted experiments varying r and fixing ℓ. 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 ℓ/r, which are similar for the values above.

As shown in Figure 13, 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 analysing 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 that called (ϵ, δ)-probabilistic differential privacy, which is a relaxation of standard differential privacy that provides ϵ-differential privacy with probability at least $1 - \delta$ [12]. 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 14 presents the results of our algorithm for such level of privacy and also for higher levels.

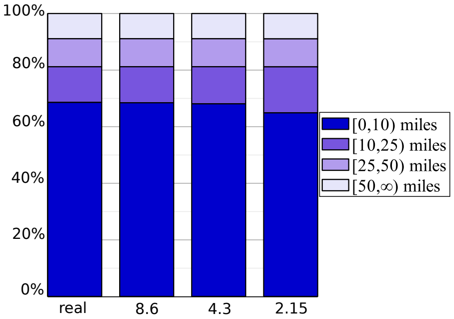

Figure 14. Home-work commute distance for $r = 3.86$, corresponding to the San Francisco county land area, and various (high) values of ℓ.

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

VII. Related Work

Much of the related work has been already discussed in Section II, here we only mention the works that were not reported there. We refer to [1] for an excellent survey on privacy methods for geolocation.

LISA [27] 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 [28] 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.

There are also some works whose main goal is to provide accurate results for data mining algorithms while preserving location privacy of the user. Gidofalvi et al. [29] use grid-based anonymization, although the privacy guarantees are mainly experimental. The method of Ho and Ruan [13] use quadtree spatial decomposition, and the concept of differential privacy [11] to develop a privacy preserving location pattern mining algorithm.

VIII. Conclusion and future work

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 with two case studies: interaction with a POI-retrieval service, and sanitization of the LODES dataset.

In the future we aim at extending our method to cope with more complex applications, possibly involving the sanitization of several (potentially related) locations. One important aspect to consider when generating noise on several data is the fact that their correlation may degrade the level of protection. We aim at devising techniques to control the possible loss of privacy and to allow the composability of our method.

TO-DO LIST

- For future work include discussion about sensible notions of *utility* for Location-Based Systems
- For future work mention the possibility of generating noise for multime locations at once (useful for the case of correlated points)
- check for consistency when refereing to side information (side knowledge , auxiliary knowledge, etc)

- The beginning of Section 2 should be shrinked or completely removed.
- When possible use "location" rather than "point"
- "small" vs "low" in section 3
- Remember to spell check the document before submitting
- Note on the metric (we discuss the euclidean, but we could also use manhattan distance)
- Maybe merge histograms in section 6 into a single figure in case we need to gain space
- Make sure that our running example is consistent throughput the paper

References

[1] M. Terrovitis, "Privacy preservation in the dissemination of location data," *SIGKDD Explorations*, vol. 13, no. 1, pp. 6–18, 2011.

[2] "Geo-indistinguishability: Differential privacy for location-based systems," Tech. Rep., 2012, **The authors are the same ones of the submitted paper. This report has been included as an attachment to our NDSS submission and should be available to the reviewers**.

[3] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proc. of MobiSys*. USENIX, 2003.

[4] B. Gedik and L. Liu, "Location privacy in mobile systems: A personalized anonymization model," in *Proc. of ICDCS*. IEEE Computer Society, 2005.

[5] M. F. Mokbel, C.-Y. Chow, and W. G. Aref, "The new casper: Query processing for location services without compromising privacy," in *Proc. of VLDB*. ACM, 2006.

[6] H. Kido, Y. Yanagisawa, and T. Satoh, "Protection of location privacy using dummies for location-based services," in *ICDE Workshops*, 2005.

[7] P. Shankar, V. Ganapathy, and L. Iftode, "Privately querying location-based services with sybilquery," in *Proc. of UbiComp*. ACM, 2009.

[8] B. Bamba, L. Liu, P. Pesti, and T. Wang, "Supporting anonymous location queries in mobile environments with privacygrid," in *Proc. of WWW*. ACM, 2008, pp. 237–246.

[9] M. Duckham and L. Kulik, "A formal model of obfuscation and negotiation for location privacy," in *Proc. of Pervasive*, ser. LNCS, vol. 3468. Springer, 2005, pp. 152–170.

[10] M. Xue, P. Kalnis, and H. K. Pung, "Location diversity: Enhanced privacy protection in location based services," in *LoCA*, ser. LNCS, vol. 5561. Springer, 2009, pp. 70–87.

[11] C. Dwork, "Differential privacy," in *Proc. of ICALP*, ser. LNCS, vol. 4052. Springer, 2006, pp. 1–12.

[12] A. Machanavajjhala, D. Kifer, J. M. Abowd, J. Gehrke, and L. Vilhuber, "Privacy: Theory meets practice on the map," in *Proc. of ICDE*. IEEE, 2008, pp. 277–286.

[13] S.-S. Ho and S. Ruan, "Differential privacy for location pattern mining," in *Proc. of Int. Workshop on Security and Privacy in GIS and LBS*. ACM, 2011.

[14] A. Khoshgozaran and C. Shahabi, "Blind evaluation of nearest neighbor queries using space transformation to preserve location privacy," in *Proc. of SSTD*, ser. LNCS, vol. 4605. Springer, 2007, pp. 239–257.

[15] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan, "Private queries in location based services: anonymizers are not necessary," in *Proc. of SIGMOD Conference*. ACM, 2008, pp. 121–132.

[16] J. Reed and B. C. Pierce, "Distance makes the types grow stronger: a calculus for differential privacy," in *Proc. of ICFP*. ACM, 2010, pp. 157–168.

[17] M. Hardt and K. Talwar, "On the geometry of differential privacy," in *Proc. of STOC*, 2010, pp. 705–714.

[18] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. S. Zemel, "Fairness through awareness," in *Proc. of ITCS*. ACM, 2012, pp. 214–226.

[19] A. Blum, K. Ligett, and A. Roth, "A learning theory approach to non-interactive database privacy," in *Proc. of STOC*. ACM, 2008, pp. 609–618.

[20] A. Roth and T. Roughgarden, "Interactive privacy via the median mechanism," in *Proc. of STOC*, 2010, pp. 765–774.

[21] C. Dwork, "A firm foundation for private data analysis," *Comm. of ACM*, vol. 54, no. 1, pp. 86–96, 2011.

[22] K. Lange and J. S. Sinsheimer, "Normal/independent distributions and their applications in robust regression," *J. of Computational and Graphical Statistics*, vol. 2, no. 2, pp. 175–198, 1993.

[23] O. Arslan, "An alternative multivariate skew laplace distribution: properties and estimation," *Statistical Papers*, vol. 51, no. 4, pp. 865–887, 2010.

[24] C. Dwork, F. Mcsherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. of TCC*. Springer, 2006, pp. 265–284.

[25] Pew Internet & American Life Project. www.pewinternet.org. [Online]. Available: www.pewinternet.org/Reports/2012/Location-based-services

[26] D. B. Rubin, "Discussion: Statistical disclosure limitation," *Journal of Official Statistics*, vol. 9, no. 2, pp. 461–468, 1993.

[27] Z. Chen, "Energy-efficient information collection and dissemination in wireless sensor networks." Ph.D. dissertation, University of Michigan, 2009.

[28] 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 *Proc. of ICDE*. IEEE, 2008.

[29] G. Gidofalvi, H. Xuegang, and T. B. Pedersen, "Privacy-preserving data mining on moving object trajectories," in *Proc. of MDM*, 2007.

Appendix

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

A. Results from Section III

Proof of Theorem 3.1: The equivalence of Geo-indistinguishability-I and III can be shown by applying Bayes' law. We show here the equivalence between Geo-indistinguishability-II and III.

Assume that K satisfies geo-indistinguishability-III. We first show that for all $r > 0$:

$$\begin{aligned}
&P(S|B_r(x))\\
&= \sum_{x'\in\mathcal{X}} P(S,x'|B_r(x))\\
&= \sum_{x'\in B_r(x)} P(S,x'|B_r(x))\\
&= \sum_{x'\in B_r(x)} P_X(x'|B_r(x))K(x')(S)\\
&\geq \sum_{x'\in B_r(x)} P_X(x'|B_r(x))e^{-\epsilon\, r}K(x)(S) \quad d(x,x')\leq r\\
&= e^{-\epsilon\, r}K(x)(S)
\end{aligned}$$

Then

$$P(x|S,B_r(x)) = \frac{P(S|x)}{P(S|B_r(x))}P(x|B_r(x)) \leq e^{\epsilon\, r}P(x|B_r(x))$$

For the opposite direction, let $x, x' \in \mathcal{X}$, let $r = d(x,x')$ and define a prior distribution $P_{X\,t}(x)$ as $P_{X\,t}(x) = t$, $P_{X\,t}(x') = 1-t$ and $P_{X\,t}(x) = 0$ for $x \neq x, x'$. Using that prior for $t \in (0,1)$ we have for all S:

$$\begin{aligned}
K(x)(S) &= P(S|x)\\
&= P(S|x,B_r(x)) \qquad\qquad x \in B_r(x)\\
&= \frac{P(x|S,B_r(x))}{P(x|B_r(x))}P(S|B_r(x))\\
&\leq e^{\epsilon\, r}P(S|B_r(x))\\
&\leq e^{\epsilon\, r}\sum_{x\in\mathcal{X}} P(S,x|B_r(x))\\
&\leq e^{\epsilon\, r}(tP(S|x) + (1-t)P(S|x'))\\
&\leq e^{\epsilon\, r}(tK(x)(S) + (1-t)K(x')(S))
\end{aligned}$$

Note that we need $t \in (0,1)$ so that $P_{X\,t}(x), P_{X\,t}(x')$ are positive and the conditional probabilities can be defined. Finally, taking the $\lim_{t\to 0}$ on both sides of the above inequality we get $K(x)(S) \leq e^{\epsilon\, r}K(x')(S)$. ■

Proof of Theorem 3.2: Let $\mathbf{x} = (x_1,\ldots,x_n), \mathbf{x}' = (x'_1,\ldots,x'_n)$ such that $d_\infty(\mathbf{x},\mathbf{x}') \leq r$. This implies that $d(x_i,x'_i) \leq r, 1 \leq i \leq n$. We have:

$$\begin{aligned}
P(\mathbf{z}|\mathbf{x}) &= \textstyle\prod_i P(z_i|x_i)\\
&\leq \textstyle\prod_i e^{\epsilon_i r}P(z_i|x'_i)\\
&= e^{r\sum_i \epsilon_i}\textstyle\prod_i P(z_i|x'_i)\\
&= e^{\epsilon r}P(\mathbf{z}|\mathbf{x}')
\end{aligned}$$

■

B. The planar laplacian satisfies geo-indistinguishability

Given the definition of $D_\epsilon(x_0)(x)$ in (2), by triangular inequality we have

$$D_\epsilon(x_0)(x) \leq e^{\epsilon\, d(x_0,x'_0)}D_\epsilon(x'_0)(x)$$

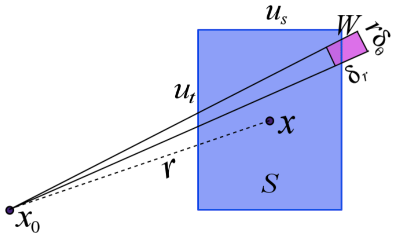

Figure 15. Bounding the probability of x in the discrete Laplacian.

Using well-known properties of integrals, we derive

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

and

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

Now, taking into account the definition of K:

$$K(x_0)(S) = \int_S D_\epsilon(x_0)(x)ds$$

we derive

$$K(x_0)(S) \leq e^{\epsilon\, d(x_0,x_0')} K(x_0')(S)$$

■

C. The discretization preserves geo-indistinguishability

Proof of Theorem 4.1: We proceed by determining an upper bound and a lover bound on $K(x_0)(x)$ for generic x_0 and x. Let $S = R(x)$. Ideally, the points remapped in x would be exactly those in S. However, due to the finite precision of the machine, the points remapped in x are those of $R_{\mathcal{W}}(x)$. Hence the probability of x is that of S plus or minus the small rectlangles[5] W of size $\delta_r \times r\,\delta_\theta$ at the border of S, see Figure 15. Let us denote by S_W the total area of these small rectangles W. Since the total perimeter of S is $2\,u_s + 2\,u_t \leq 4\,u$, we have that $S_W \leq 4\,u\,\delta_r$. The probability density on this area is at most $({\epsilon^2}/{2\pi})e^{-\epsilon(r-u/\sqrt{2})}$, where $r = d(x_0, x)$. Summarizing:

$$\int_S D_\epsilon(x_0)(x_1)ds - P(r) \leq K(x_0)(x) \leq \int_S D_\epsilon(x_0)(x_1)ds + P(r) \qquad (4)$$

where $P(r) = 4\,u\,\delta_r({\epsilon^2}/{2\pi})e^{-\epsilon(r-u/\sqrt{2})}$. 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

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

from which we derive

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

from which, using (4), we obtain

$$K(x_0)(x) \leq e^{\epsilon\, d(x_0,x_0')}(K(x_0')(x) + P(r')) + P(r) \qquad (5)$$

where $r' = d(x_0', x)$. Finally, for $d(x_0', x) \leq r_{\max}$ we have:

$$e^{\epsilon\, d(x_0,x_0')}(K(x_0')(x) + P(r')) + P(r) \leq e^{\epsilon'\, d(x_0,x_0')} K(x_0')(x),$$

which, together with (5), concludes the proof. ■

D. The truncation preserves geo-indistinguishability

Proof of Theorem 4.2: The proof proceeds like the one for Theorem 4.1, 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 $R(x) \setminus S_W$, but also by the probability of the part C of the cone determined by o, $R(x)$, and lying outside $\mathcal{A}$ (see Figure **??**). Following a similar reasoning as in the proof of Theorem 4.1 we get

$$\int_{S\cup C} D_\epsilon(x_0)(x_1)ds - P'(r) \leq K^T(x_0)(x)$$

and

$$K^T(x_0)(x) \leq \int_{S\cup C} D_\epsilon(x_0)(x_1)ds + P'(r)$$

where $P'(r) = 2\,u\,\delta_r({\epsilon^2}/{2\pi})e^{-\epsilon(r-u/\sqrt{2})}$. The rest follows as in the proof of Theorem 4.1. ■

[5] W is actually a fragment of a circular crown, but when δ_θ is very small, it approximates a rectangle.