Location Privacy and Attacker Knowledge: Who Are We Fighting Against?

Rinku Dewri

Department of Computer Science
University of Denver, Denver CO 80208, USA
rdewri@cs.du.edu

Abstract. Location privacy research has received wide attention in the past few years owing to the growing popularity of location-based applications, and the skepticism thereof on the collection of location information. A large section of this research is directed towards mechanisms based on location obfuscation. The primary motivation for this engagement comes from the relatively well researched area of database privacy. Researchers in this sibling domain have indicated multiple times that any notion of privacy is incomplete without explicit statements on the capabilities of an attacker. The question we ask in the context of location privacy is whether the attacker we are fighting against exists or not. In this paper, we provide a classification of attacker knowledge, and explore what implication does a certain form of knowledge has on location privacy. We argue that the use of cloaking regions can adversely impact the preservation of privacy in the presence of approximate location knowledge, and demonstrate how perturbation based mechanisms can instead be useful.

Key words: location privacy, differential privacy, query approximations

1 Introduction

Location based applications are geared towards providing services tailored to the current location of a user. These applications utilize the positioning capabilities of a mobile device to determine the current location of the user, and customize query results to include neighboring points of interests. Wide acceptance of personal digital assistants and the advancements in wireless cellular technology have opened up countless possibilities in this business paradigm. Potential applications can range from proximity based notifications to tracking business resources. A wireless carrier typically serves as a channel between the user and the location content provider.

The potential advantages of location based applications is not difficult to realize. However, location knowledge is often perceived as personal information. It remains an open question whether the benefits of these applications can outweigh the underlying privacy risks. A similar question has been around for more than a decade in the field of database privacy. Databases hosting our personal information can serve as data mining grounds to facilitate research studies in a variety of fields. At the same time, the same information in the hands of an adversary can have alarming ramifications. Database privacy preservation is an ongoing effort to design data sharing methods in order to prevent such an adversary from making personal inferences using the shared data [1, 2, 3]. Drawing inspiration from these efforts, location based applications have been argued to be usable without communicating precise location data to the content provider.

Location obfuscation is a widely researched technique to achieve location privacy. The fundamental idea here is to process location based queries relative to a sufficiently larger region, also known as a cloaking region, compared to one where a user can be uniquely located. For instance, a cloaking region can be generated to include k users, including the one making the query [4]. Multiple algorithms have been proposed to generate such a k-anonymous cloaking region [5, 6]. However, as demonstrated in the case of database privacy, obfuscating private data without understanding the capabilities of the attacker can be unproductive [7, 8, 9]. A privacy preserving mechanism is not better or worse than another. It is the adversary who is weaker or stronger. The background knowledge of the attacker must be known (or at least assumed) in order to demonstrate the privacy guarantees of a mechanism.

We begin this work by identifying the primary form of attacker knowledge targeted by most location obfuscation techniques. This knowledge relates to an attacker being able to determine the true locations of a certain subset of users. Using a case by case analysis of what this attacker can achieve from queries made using true locations and queries made using cloaking regions, we argue that "location privacy" is a misused term in this context. The use of cloaking regions is motivated by the need to introduce ambiguity in correlating a user to a query. However, if an attacker does not have any location knowledge of the users, then location information in a query cannot be used to map it to a user. The attacker must posses at least approximate location knowledge about the user, to be able to exploit the location information in a query. On the other hand, if true location knowledge is present, then there is no location privacy. In fact, what is being offered is query privacy. We treat the two forms of privacy differently – location privacy meaning hiding the location and query privacy meaning preventing the mapping of a query to a user.

We also justify that cloaking regions are insufficient in preserving privacy when an attacker has approximate location knowledge. Although cloaking regions do not directly disclose the true locations, we believe that no privacy mechanism should enable an attacker to improve upon the existing background knowledge. The knowledge gain should be formally bounded in the worst case. Towards this end, we explore the possibility of using perturbed locations to issue queries and propose a perturbation method based on differential privacy [10]. Differential privacy works under the principle that the chances of being a victim of a privacy breach should not increase substantially due to the inclusion of ones private information in a shared data set. The perturbed location is differentially k-anonymous, in the sense that the probability ratio of any two of the k users is bounded. Empirical results are provided to demonstrate that such queries can retrieve a significantly large subset of the actual query results.

The remainder of the paper is organized as follows. Section 2 initiates our discussion on attacker capabilities, and the affect on location and query privacy. Section 3 presents our approach to address a form of attacker knowledge based on approximate locations of the users. Section 4 presents some empirical results on the effectiveness of the approach in generating useful query results. Section 5 lists some related work in this area, followed by references to future work in Section 6.

2 Attacker Class

Classification of attacker knowledge is crucial in order to provide a comprehensive statement on the privacy preserving properties of an obfuscation technique. To consider the extremes, location obfuscation in the presence of an "oracle" attacker, or an attacker with effectively no background knowledge, is only going to degrade the quality of service. Other intermediate scenarios also exist where location obfuscation cannot achieve one or both of location and query privacy. We begin with two forms of background knowledge that an attacker is likely to have.

The first form of background knowledge is related to the location of users. An adversary that has information on the locations of any individual(s) is referred to here as a *locator*. Further, a *perfect* locator knows exact coordinates of the users, while an *approximate* locator has approximate knowledge (an area instead of exact coordinates) on the locations. The second form of knowledge is related to the identity of users issuing the queries. We refer to any adversary that has access to the query database as a *holder*. A *perfect* holder in this case would be an adversary who knows the identity of the person who issued a query.

There are multiple permutations in which these two forms of knowledge may be present in an adversary. While each form in itself states how much an attacker knows about the locations or queries of the users, respectively, the objective is to avoid the inference or improvement of one form of knowledge using existing knowledge of the other form. Hence, given a certain level of background knowledge, we consider a privacy breach to have occurred if and only if the adversary gains additional knowledge. Gaining additional knowledge in this case refers to instances such as a perfect locator becoming a perfect holder (and vice versa), or an approximate locator improving its location approximations.

Location based service users communicate location information as part their queries. The location information can be in the form of precise GPS coordinates, the resulting query being processed thereafter with respect to a point in space. Such queries are also referred to as *point queries*. However, due to the implications on privacy, precise locations are obfuscated using a cloaking region. Queries in this case are processed on a geographic range, therefore referred to as *range queries*. We begin with point queries and put the two forms of attacker knowledge in perspective with respect to such queries. Some of the observations in the following section are well-known in the community. We present them here for the sake of completeness.

2.1 Point Queries

A point query is where exact geographic coordinates are communicated along with the query. A query database in this case contains the precise location of users, among other parameters of the queries. It is a straightforward observation that no location privacy can be achieved in the presence of a perfect locator, and no query privacy can be achieved in the presence of a perfect holder. Nonetheless, query privacy is preserved in the case of a perfect locator. However, as an immediate consequence of point queries, location privacy is violated even when the adversary is only a perfect holder. A perfect holder in this case performs a identity to location mapping using the location information in the query database. A perfect locator must also be at least a holder to effectuate a breach of query privacy. In this case, the adversary uses the location knowledge to determine the corresponding query of the user in the database. The perfect locator here covers situations such as restricted space identification and observation based identification [4]. A simple holder with access to the query database alone is no threat to either location or query privacy of the users.

The effectiveness of point queries in the presence of approximate locators has not been evaluated yet. Point queries can be potentially harmless depending on the extent of the adversary's approximation. For instance, an approximate locator with an approximation of a few hundred meters is stronger than one with an approximation of a city block. The exact extent of knowledge is difficult to estimate. We shall discuss later how point queries can still be effectively generated in the presence of approximate locators.

2.2 Range Queries

A range query is where a query region is associated with the query. Query results are generated assuming that the user may be located anywhere inside the region. The query region serves as a cloak for the user, and is generated following some established privacy principle. For instance, a k-anonymous cloaking region would encompass at least k users inside it. Large cloaking regions would potentially result in the communication of a larger result set and degrade the QoS levels of the system. Hence, the obfuscation algorithm tries to achieve the privacy principle within the smallest possible area. In the following, we present a case by case overview of which privacy aspect does a range query help preserve, and under what form of adversarial knowledge.

Perfect Locator Since a perfect locator knows the location of a user, use of a cloaking region does not help hide the location of the user. Query privacy is preserved in the absence of access to the query database. This implies that no privacy breach (in the sense of gaining additional knowledge) can occur in the presence of this type of adversary. Point queries can in fact be used instead of a range query, in order to improve the quality of service.

Approximate Locator Cloaking regions also do not help achieve better location privacy from an approximate locator. The approximation of the adversary on the user's location is what determines the location privacy level. Point queries can again be used here, given that the adversary has no access to the query database. In other words, no location privacy violation can occur as a side-effect of the user using the service.

Perfect Holder No query privacy is possible in the presence of a perfect holder. Location privacy violation is certain since the cloaking regions present in the queries provide approximate location knowledge to the adversary. The cloaking regions can potentially reveal more precise information as well. Note that a privacy principle such as k-anonymity is meant to prevent the association of a user to the issued query – any of the k users could have issued the query. However, such a principle is irrelevant in the case of a perfect holder. A better principle to enforce would be location diversity [11, 12]. This would guarantee that zones with multiple levels of sensitivity are present within the cloaking region, thereby preventing further location based inferences. Request locality is another issue to address. This situation occurs when different likelihoods can be estimated for issuing the query from different areas within the cloaking region.

Holder A simple holder with no location based knowledge is unable to correlate a cloaking region to a specific user. Both location and query privacy are preserved. This is the weakest form of an adversary. Note that an adversary who is not a perfect or approximate locator cannot determine if a user is inside a cloaking region. Hence, a range query might as well be replaced with a point query.

Perfect Locator and Perfect Holder As in the case of point queries, no level of obfuscation can hide the location and query of a user from this form of an adversary.

Perfect Locator and Holder The location of a user is already known to this kind of an adversary. It is easy to determine the set of queries that could have potentially originated from a certain user. However, query privacy violation can be prevented if the cloaking region can generate an ambiguous mapping between a query and the user. This is achieved by anonymity principles such as k-anonymity. In fact, obfuscation methods that generate minimal k-anonymous cloaking regions assume the existence of a perfect locator with precise location knowledge of at least k users. This assumption implies that location obfuscation is used here to preserve query privacy, and not necessarily any form of location privacy. Query privacy, however, can also be preserved by issuing a point query using the true location of one of the k users. This can produce a relatively accurate result set if the bounding rectangle of the k users is not excessively large. The result sets would differ much for larger bounding rectangles, in which case the communication costs may itself be too high for acceptable range query processing.

Approximate Locator and Perfect Holder Approximate locators have the ability to correlate a user with a geographic region. The size of this region is not constant, and is an attribute related to the adversary's background knowledge. Under such a scenario, it becomes difficult to create a cloaking region that encompasses the entire area within the locator's approximation. Hence, it is possible that a perfect holder uses the cloaking region in a query to narrow down the geographic region where the user is located. Cloaking regions can therefore provide additional location knowledge to an adversary, thereby leading to a location privacy breach.

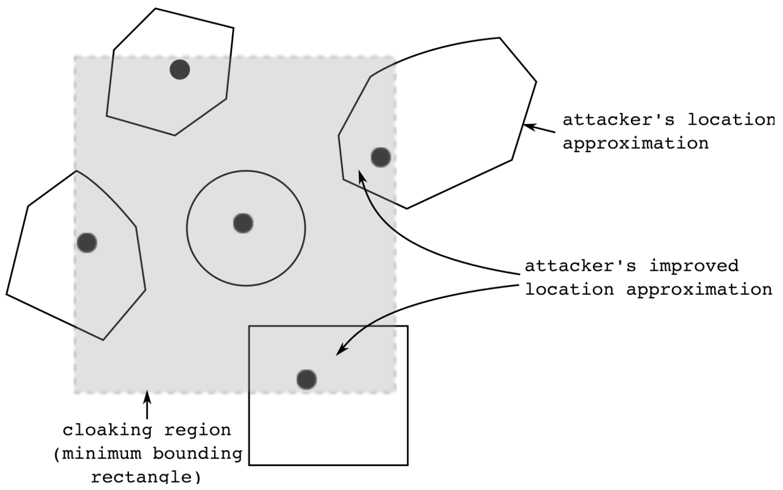
attacker's location approximation
attacker's improved location approximation
cloaking region (minimum bounding rectangle)

Fig. 1. Location privacy breach as a result of using cloaking regions.

Approximate Locator and Holder While cloaking regions are sufficient (although perhaps not always required) to handle a perfect locator and holder, their use starts to have a detrimental affect in the presence of approximate locators. As depicted in Fig. 1, a k-anonymous cloaking region may allow an approximate locator to improve upon the location knowledge of more than just the query issuer. The problem is eliminated only if the cloaking region is guaranteed to encompass the approximated regions corresponding to each of the k users. Unfortunately, it is difficult to judge the extent of knowledge that an adversary possesses. This case presents us with a situation where the obfuscation method helps preserve query privacy but can potentially lead to a breach in location privacy.

Note that most privacy preservation attempts address perfect locators and holders. Therefore, the term "location privacy" seems to have been misused, in the sense that true location knowledge is already assumed to be known to the adversary. Query privacy is a more appropriate term to use in this context.

We summarize below the conclusions that can be drawn from the discussion in the preceding sections.

1. Neither location privacy nor query privacy can be preserved in the presence of a perfect locator and a perfect holder.
2. Point queries pose privacy threats in the presence of a perfect locator and holder.
3. Cloaking regions help preserve query privacy in the presence of a perfect locator and holder.
4. Range queries may be replaceable by point queries in the above case.
5. Cloaking regions can provide query privacy in the presence of an approximate locator and holder, but do not guarantee protection against a location privacy breach.

3 Approximate Locators

Current location obfuscation techniques based on cloaking regions are insufficient, and undesirable, in location privacy preservation. This arises from the fact that perfect locators represent a very strong class of attackers. For instance, acquiring the exact geographic coordinates of a user would require satellite based monitoring capabilities. Further, not much can be done with location obfuscation once an adversary gains access to such information. A more plausible form of adversary is represented by an approximate locator. Approximate location knowledge can be obtained by a variety of means – device communication logs such as cell towers used, public records such as parking violations, or social engineering methods such as a "water-cooler conversation." Preserving location privacy in this context dwells upon the problem of preventing an attacker from reducing the margin of location error using external references of a user's activities (such as in a location based service log).

Recall that cloaking regions are insufficient in providing location privacy against an approximate attacker. Hence, we revert back to point queries and analyze if they can be used in a manner that preserves location privacy. Perturbation of user locations is the basis of this analysis. An attacker can identify common areas between a cloaking region and an approximate location in order to improve the approximation. This is possible because cloaking regions always cover the true location of the querying user (amongst others). However, a perturbed location is a single point in space that could have been generated by any user in a given set.

Queries based on perturbed locations can result in an inaccurate result set. However, if the perturbations are reasonably close to the actual location, then the query results can also be assumed to be close enough to the true set. There is definitely an inherent trade-off involved between the accuracy of the result set and the location perturbations. We postpone the analysis of this trade-off for a later stage and focus on the generation of the perturbations themselves.

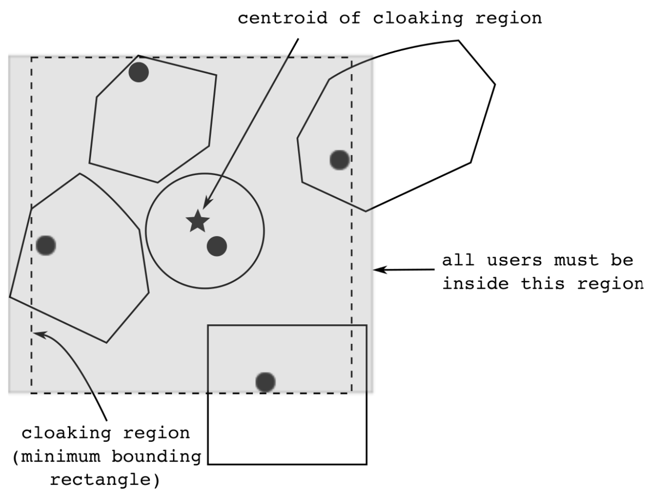
centroid of cloaking region
all users must be inside this region
cloaking region (minimum bounding rectangle)

Fig. 2. Improving location approximations using the centroid of the cloaking region.

A trivial method to perturb a user's location would be to use the centroid of a k-anonymous cloaking region while issuing the query. An adversary with exact location information can employ an inversion attack to determine the set of k locations used to arrive at the perturbed coordinates. An inversion attack would involve re-computing the centroid of the bounding rectangle derived from different sets of k location coordinates, with the objective of matching the coordinate in the query. Note that we do not consider any privacy parameter (including k) to be hidden from the attacker. Executing an inversion attack is not straightforward for an approximate locator. Depending on the size of the k-anonymous cloaking region, the centroid can also serve as a good estimate of a user's location and possibly generate a significantly accurate set of results. However, as depicted in Fig. 2, owing to the equi-distance property of the centroid, its ability to prevent a location privacy breach is still questionable. In the figure, the grey area bounded by the solid line represents the largest possible region that can be a bounding rectangle (users can be anywhere in the approximate regions) and has a centroid same as the true minimum bounding rectangle. This improves the location approximation corresponding to two of the users. The applicability of other notions of a centroid remains to be explored.

Our approach is motivated by the requirement to provide probabilistic bounds on what an adversary can learn from the perturbed location. We adopt the differential privacy approach in statistical databases in this context [10].

3.1 Location Perturbation

Let l_p be the perturbed location corresponding to a true location l_t, denoted as $l_t \rightarrow l_p$. A location is assumed to have two components, denoted by the non-negative x and y coordinates. Let $l_1, ..., l_k$ be a set of k points, one of which is

l_t. The method of choosing these k points is discussed in the next section. We would generate the perturbed location $l_p = (x_p, y_p)$ such that

$$Pr(x_i \to x_p) \leq e^{\epsilon} Pr(x_j \to x_p) \text{ and}$$

$$Pr(y_i \to y_p) \leq e^{\epsilon} Pr(y_j \to y_p)$$

where $\epsilon \geq 0$ and $i, j \in \{1, ..., k\}$. We achieve this property by using a Laplace distribution with scale $\lambda > 0$ to perturb a location $l_i = (x_i, y_i)$ such that

$$Pr(x_i \to x_p) = \frac{1}{2\lambda} e^{-\frac{|x_i - x_p|}{\lambda}} \text{ and}$$

$$Pr(y_i \to y_p) = \frac{1}{2\lambda} e^{-\frac{|y_i - y_p|}{\lambda}}.$$

Based on the following observation, λ is set at $(\max_n x_n - \min_n x_n)/\epsilon$ to generate x_p, and set at $(\max_n y_n - \min_n y_n)/\epsilon$ to generate y_p. l_p is obtained as (x_p, y_p).

Observation: Without loss of generality, let c denote a generic component of a location. Using the triangle inequality, we can write $|c_j - c_p| \leq |c_j - c_i| + |c_i - c_p|$. After rearrangement, dividing by λ, raising as a power of e and multiplying by $1/2\lambda$, we get

$$\frac{1}{2\lambda} e^{-\frac{|c_i - c_p|}{\lambda}} \leq \frac{1}{2\lambda} e^{-\frac{|c_j - c_p|}{\lambda}} e^{\frac{|c_j - c_i|}{\lambda}} \text{ , or}$$
$$Pr(c_i \to c_p) \leq Pr(c_j \to c_p) e^{\frac{|c_j - c_i|}{\lambda}}.$$

We therefore have

$$Pr(x_i \to x_p) \leq Pr(x_j \to x_p) e^{\frac{|x_j - x_i|}{\lambda}} \text{ and}$$
$$Pr(y_i \to y_p) \leq Pr(y_j \to y_p) e^{\frac{|y_j - y_i|}{\lambda}},$$

and the power of the exponent is bounded as

$$Pr(x_i \to x_p) \leq Pr(x_j \to x_p) e^{\frac{\max_n x_n - \min_n x_n}{\lambda}} \text{ and}$$
$$Pr(y_i \to y_p) \leq Pr(y_j \to y_p) e^{\frac{\max_n y_n - \min_n y_n}{\lambda}}.$$

Using the Laplace distributed noise also ensures that

$$Pr(c_i \to c_p) \geq e^{-\epsilon} Pr(c_j \to c_p).$$

The following inequalities verify that the desired property can be achieved for any component c in l_i and l_j.

$$e^{-\epsilon} \leq \frac{Pr(c_i \to c_p)}{Pr(c_j \to c_p)} \leq e^{\epsilon}$$
$$\Longleftrightarrow e^{\epsilon} \geq \frac{Pr(c_j \to c_p)}{Pr(c_i \to c_p)} \geq e^{-\epsilon} \text{ with } \epsilon \geq 0.$$

Hence, the probability of a location coordinate generating a certain perturbed value is always within a factor e^{ϵ} of the probability of some other location (in the set of k points) generating the same perturbed value. In the k-anonymity sense, any of the k points could have been used to generate the perturbed location.

3.2 Selecting a Perturbation

A perturbed location for a query point can be chosen using the above method. However, the distribution of the k points can affect the proximity of the perturbed location to the true coordinates. Further, the k points should be chosen to preserve reciprocity [6, 13]. In other words, the same set should be chosen irrespective of which of the k locations is the query point. This is achieved by dividing the users into buckets of size k, the set being chosen as the bucket to which the query point belongs. Each of the k points is subjected to perturbation, and the one having the minimum average distance to all points in the set is chosen as the location to issue the query. Given a perturbed location, the k points are probabilistically identical (within a factor of e^{ϵ}) irrespective of which one was used to perform the perturbation. Hence, choosing the one with minimum average distance to all points does not risk an inversion attack. Note that the context of the application still plays a crucial role. If the user base is relatively sparse, i.e. the k users are distributed over a significantly large area, then the generated perturbation will be far away from the true location. A cloaking region could also be unacceptably large in this case.

Algorithm 1 lists the pseudo code of the approach. The function returns a perturbed location of a user $\mathcal{U}$. Lines 1 to 11 determine the k size bucket to which the user belongs. The buckets are formed based on the Hilbert indices of the users. The locality preserving properties of Hilbert curves ensure (although not necessarily optimal) the formation of buckets with users that are at close proximity to each other. Error checks and boundary conditions are not shown in the code. For instance, if a user belongs to the last bucket and its size is less than k, then the last bucket should be merged with the previous one. Lines 14 to 17 compute a perturbed value corresponding to the location of every user in the bucket. Each component c of a location is perturbed to $c - \lambda \text{sign}(rnd) \ln(1 - 2|rnd|)$, where rnd is a random value between -0.5 and 0.5 drawn from a uniform distribution, and λ is set as described in the previous section. This makes the perturbation Laplace distributed around c.

3.3 Evaluating the Perturbation

Cloaking regions guarantee that the results generated for a location based query will contain the results corresponding to the location of the user. Such a claim

Algorithm 1 Location Perturbation

```
Require: User $\mathcal{U}$ with associated $k$.
Ensure: A perturbed location for $\mathcal{U}$.
 1: $\mathcal{H}$ = set of all users sorted by their Hilbert index
 2: repeat
 3:   $\mathcal{D} = \phi$
 4:   for all ($u \in \mathcal{H}$ in order) do
 5:     $\mathcal{D} = \mathcal{D} \cup \{u\}$
 6:     if ($|\mathcal{D}| = k$) then
 7:       break
 8:     end if
 9:   end for
10:   $\mathcal{H} = \mathcal{H} - \mathcal{D}$
11: until ($\mathcal{U} \in \mathcal{D}$)
12: $\mathcal{L} = \{$location of $u \in \mathcal{D}\}$
13: $\mathcal{L}_p = \phi$
14: for all ($l \in \mathcal{L}$) do
15:   $l_p$ =perturbed $l$
16:   $\mathcal{L}_p = \mathcal{L}_p \cup \{l_p\}$
17: end for
18: return $l_p \in \mathcal{L}_p$ such that $l_p$ has minimum average distance from $\mathcal{L}$
```

cannot be made for queries issued with a perturbed location. However, it remains to be evaluated how different is the result set when generated with respect to the true location, compared to that generated with respect to a perturbed location. Differences in the result may or may not exist depending on the density of the queried objects, and the distance of the perturbed location from the true one. A Knn-query, for instance, on sparsely distributed objects (e.g. hospitals) is likely to generate a larger subset of common results. On the other hand, for densely distributed objects, this likelihood reduces. K here is the number of nearest neighbor objects to retrieve corresponding to a location. Note that we use a lower case k for the computation of a perturbed location.

Result set similarity can also be measured with respect to the distances to the retrieved objects. Under this measure, two result sets are considered similar if, corresponding to every object in one set, there exists an object in the other set that is equi-distant from the queried location. This perspective of result similarity applies well to proximity based queries – nearest gas stations, nearest restaurants, nearest friends – where the distance to the object carries more weight than attributes of the objects. Result set similarity using common subsets is relevant in queries where the retrieved objects must be ordered using user-stated preferences – nearest K cheapest gas stations.

A third measure is also possible using the distance of the perturbed location from the true location. Assuming that the service provider guarantees that the result set is accurate relative to the query point, a user wanting complete accuracy will have to travel from the current location to the perturbed point. It is therefore worth investigating how far is the generated perturbation from the current location of the user.

Although we are not stating any theoretical bounds on these metrics at this stage, intuition says that query processing relative to well-formed perturbed locations will not be futile. As the first step, the following three metrics are used to evaluate the effectiveness of our approach [14].

1. *Nearness:* Fraction of perturbations at close proximity to the true location.
2. *Displacement:* Let $\mathcal{O} = \{o_1, ..., o_K\}$ be the objects retrieved by a Knn-query relative to the true location of user $\mathcal{U}$, and $\mathcal{O}' = \{o'_1, ..., o'_K\}$ be the objects retrieved relative to the perturbed location. The displacement is then given as
$$\sum_{i=1}^{K} dist(o'_i, \mathcal{U}) - \sum_{i=1}^{K} dist(o_i, \mathcal{U}),$$
$dist(\cdot)$ being a distance function. The minimum possible displacement is zero.
3. *Resemblance:* Fraction of common objects between $\mathcal{O}$ and $\mathcal{O}'$, given as
$$\frac{|\mathcal{O} \cap \mathcal{O}'|}{|\mathcal{O}|}.$$

4 Empirical Results

We have generated a trace data set using a simulator that operates multiple mobile objects based on real-world road network information available from the National Mapping Division of the US Geological Survey. We use an area of approximately 168 km^2 in the Chamblee region of Georgia, USA for this study (Fig. 3). Three road types are identified based on the available data – expressway, arterial and collector. Real traffic volume data is used to determine the number of users on the different road types [4]. The total number of users on a road type vary proportionately to the total length and traffic volume of the road type, and reciprocally to the average speed. The mean speed, standard deviation and traffic volumes on the road types are shown in the figure. Using the number of users on each road type, the simulator randomly places them on the network and moves them around. The users move with a speed drawn from a normal distribution, randomly making turns and changing speed at junctions. The simulator maintains the traffic volume statistics while moving the users.

The used traffic volume information results in 8,558 users with 34% on expressways, 8% on arterial roads and 58% on collector roads. The trace data consists of multiple records spanning one hour of simulated time. A record is made up of a time stamp, user identifier, and x and y coordinates of the user's location. The granularity of the data is maintained such that the Euclidean distance between successive locations of the same user is approximately 100 meters. Each user has an associated k value drawn from the range $[2, 50]$ by using a Zipf distribution favoring higher values and with the exponent 0.6. The trace data is sorted by the time stamp of records. The first minute of records is used for

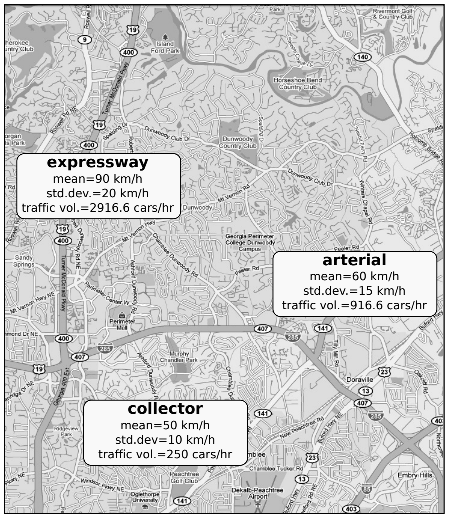

Fig. 3. Simulation performed over an area of Chamblee, GA, USA.

initialization. Location coordinates in each record thereafter are subjected to perturbation. Over 4,000,000 records are processed during a pass of the trace data.

Queried objects are distributed randomly over the entire map based on a density value (number of objects per km^2). A Knn-query is issued relative to every perturbed location. Displacement is measured using a Euclidean distance metric. The entire map is assumed to be on a grid of $2^{14} \times 2^{14}$ cells (a cell at every meter) while calculating the Hilbert indices [15]. Objects in the same cell have the same Hilbert index. All simulation results are obtained on a 2.8GHz Quad-Core Intel Xeon machine with 8GB memory and running Mac OSX 10.6.7.

Fig. 4 shows the number of perturbations that resulted in the perturbed point being generated within 5000 meters of the user's actual location. A value of $\epsilon = 0.01$ effectuates to saying that two users should effectively have the same probability of generating the perturbation ($e^\epsilon = 1.01$). This is difficult to achieve for most values of k. As the ϵ value approaches 0.5 ($e^{0.5} = 1.65$), we see a useful distribution. At this point, more than 90% of the perturbations are within 1000 meters of the true location (Table 1). 60% of the points are in a much closer proximity of 500 meters. The numbers increase favorably with increasing ϵ. However, higher values of ϵ reduce the practical significance of the approach. For instance, with $\epsilon = 2.0$, we are already willing to accept a factor of 7 difference in the probability estimates. Nonetheless, it is promising to see that significantly high nearness values are possible with smaller values of ϵ as well.

Fig. 5 shows the resemblance and displacement values corresponding to different values of K (the number of nearest neighbors to retrieve) and density. The values are averaged over the the total number of requests processed (4484683). A density of 0.1 results in 25 objects across the entire region (sparsely distributed),

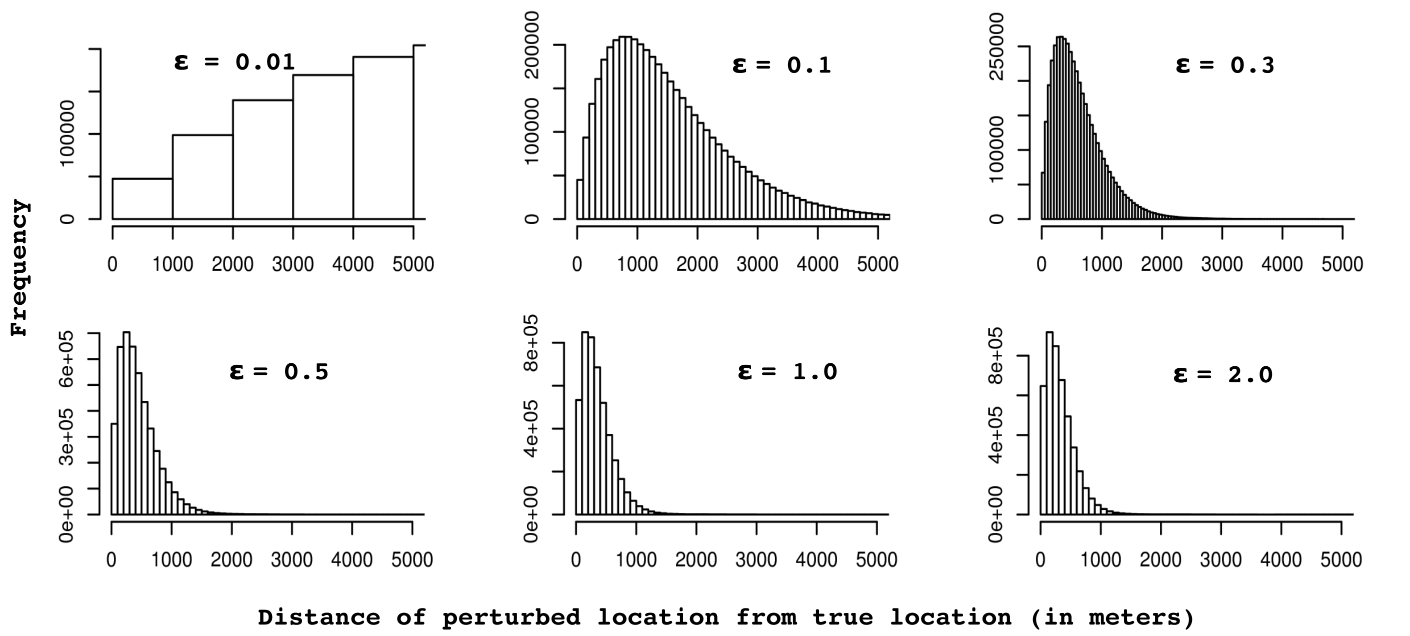

Fig. 4. Number of anonymization attempts where the perturbed location is within 5000 meters of the true location. Perturbed location is within 1000 meters for more than 90% of the attempts with $\epsilon = 0.5$. Total number of anonymization attempts = 4484683.

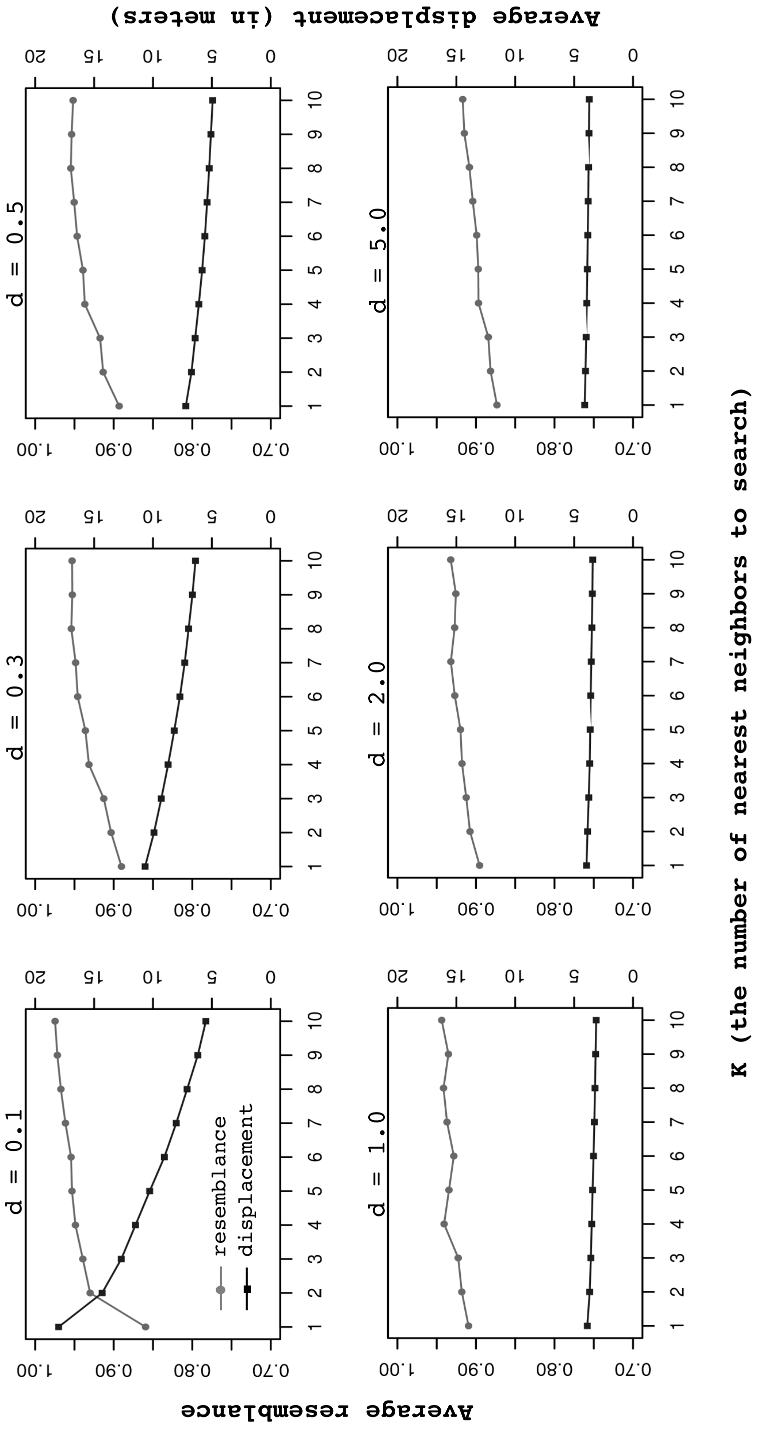

Fig. 5. Average resemblance and displacement values for *Knn*-queries on objects distributed with various density *d*. Perturbations are generated with $\epsilon = 0.5$.

Table 1. Percentage of anonymization attempts where perturbed location is at close proximity to true location.

ϵ	$\leq 1000m$	$\leq 500m$	$\leq 100m$
0.01	1.05	0.37	0.01
0.1	36.61	13.70	1.00
0.3	84.16	48.33	4.64
0.5	93.79	64.53	7.81
1.0	97.41	76.10	11.89
2.0	98.11	79.91	14.42

while a value such as 5 results in 980 objects (densely distributed). Subset similarity (resemblance) is over 80% on the average. However, the metric shows a slow decreasing trend as objects become more densely situated. The chances of finding points of interest in the neighborhood increases as they become more closely packed. The displacement is still minimal in this case. Differences in the distance are within a mere 5 meters for the simulated objects. Query results on sparse objects can be comparatively distant, but still acceptable. The number of objects to retrieve has an influence in this case. While the resemblance values are more or less similar, displacement is comparatively higher when a smaller number of objects are retrieved on sparse objects. A nearest neighbor search ($K = 1$) still retrieves the same object on more than 80% of the queries processed.

5 Related Work

Location obfuscation has been earlier achieved either through the use of dummy queries or cloaking regions. In the dummy query method, a user hides its actual query (with the true location) amongst a set of additional queries with incorrect locations [16, 17]. The user's actual location is one amongst the locations in the query set. Using false dummies affect query privacy if user locations are known to the attacker. Cheng et al. propose a data model to augment uncertainty to location data using circular regions around all objects [18]. They use imprecise queries that hide the location of the query issuer and yield probabilistic results, modeled as the amount of overlap between the query range and circular region around the queried objects. Yiu et al. propose an incremental nearest neighbor processing algorithm to retrieve query results [19]. The process starts with an anchor, a location different from that of the user, and it proceeds until an accurate query result can be reported. Trusted third party based approaches rely on an anonymizer that creates spatial regions to hide the true location of users. The anonymizer communicates this region to the content provider and then filters the result set accordingly. Gedik and Liu develop a location privacy architecture where each user can specify a minimum anonymity level, and maximum temporal and spatial tolerances while creating the cloaking regions [5]. Ghinita et al. propose a decentralized architecture to construct an anonymous spatial region, and eliminate the need for the centralized anonymizer [20]. Kalnis et al.

propose that all obfuscation methods should satisfy the reciprocity property [6]. This prevents inversion attacks where knowledge of the underlying anonymizing algorithm can be used to identify the actual object. Mokbel et al. explore query processing of different types on spatial regions – private queries over public data, public queries over private data, and private queries over private data [21]. Lee et al. explore privacy concerns in path queries where source and destination inputs may reveal personal information about users [22]. They propose the notion of obfuscated path queries where multiple sources and destinations are specified to hide the true inputs. Xu and Cai argue that the impact of a privacy parameter, such as k, on the level of privacy is often difficult to perceive. They treat privacy as a feeling-based property and propose using the popularity of a public region as the privacy level [23]. Each user specifies a spatial region as its privacy index, and the cloaking region for the user must at least have the same popularity as that of the specified region. An entropy based computation is used to define the popularity of a spatial region. Soriano et al. show that the privacy assurances of this model do not hold when the adversary possesses footprint knowledge on the spatial regions over time [24]. Shokri et al. propose a framework to quantify location privacy based on the expected estimation error of an adversary [25].

Data transformation is another method to prevent the inference of locations. Agrawal et al. propose an encryption technique called OPES (Order Preserving Encryption Scheme) that allows comparison operations to be directly applied on encrypted data [26]. Operand decryption is however required for computing SUM and AVG. Wong et al. overcome this drawback by developing an asymmetric scalar-product preserving encryption [27]. This allows the preservation of relative distances between database points. Khoshgozaran et al. employ Hilbert curves to transform the data points and then answer queries in the transformed space [14]. The parameters of the transformation, called the Space Decryption Key, is assumed to be not known to an adversary. A new paradigm in location privacy is based on private information retrieval (PIR) techniques. Khoshgozaran et al. propose K nearest neighbor queries that can be reduced to a set of PIR block retrievals [28]. These retrievals can be performed using a tamper-resistant processor located at the server so that the content provider is oblivious of the retrieved blocks. Papadopoulos et al. further warrant the need to retrieve the same number of blocks across queries [29].

6 Conclusions

Obfuscated locations can provide the means to access a location based service without risking privacy breaches. The strength of the obfuscation itself is dependent on the background knowledge of the attacker. Cloaking regions can be used to provide query privacy, but at the same time, can also enable an attacker with approximate location knowledge to improve its approximations. We propose a method based on location perturbation to address such attackers. Perturbed locations are generated using a Laplace distributed noise function in a way such that any user, from a set of k users, is likely to be the query issuer within a parameterized bound. Empirical evaluation shows that the perturbed locations can still serve as promising query points. A high fraction of the actual result set can be retrieved, or otherwise, similarity in distances to the points of interest can be achieved.

Resolution of bad perturbations is an issue that remains to be explored. These are perturbations that are significantly far away from the true locations. While their occurrence has not been found to be very high in the empirical study, it needs to be determined if they can be eliminated altogether. Reducing the value of k may have a positive impact, but at the expense of reduced anonymity. In addition, the k value is only used to determine a set of close neighbors that can be used to compute a noise level for the perturbations. Its may be possible to adaptively choose the value based on the proximity of the perturbations to the true locations. Further, the result set similarity could be improved upon by using queries from multiple perturbed locations. Decentralized computation of the perturbations should not be difficult, given a framework to determine the k users.

We have not considered another possible form of adversarial knowledge in this study. These adversaries, called *crossholders*, posses knowledge on the identity of individuals who did not issue a certain query. Consequently, a k-anonymous cloaking region in this case is $(k-n)$-anonymous, where n is the number of individuals that the adversary can eliminate. k-anonymity can still be achieved by ensuring the cloaking region is $(k+n)$-anonymous. As in the case of approximate locators, the difficulty lies in determining the attacker's extent of knowledge – the value of n. The perturbation based approach demonstrated here is also weak against such adversaries, specifically because of the underlying usage of k-anonymity. The dependence on k can be removed by using the maximum L_1-norm distance between all users in the variance computation. However, such high levels of variance can make the perturbed locations significantly distant from the true locations, and effectively useless in generating relevant results.

References

1. LeFevre, K., DeWitt, D.J., Ramakrishnan, R.: Incognito: Efficient Full-Domain k-Anonymity. In: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, pp. 49–60. (2005)
2. LeFevre, K., DeWitt, D.J., Ramakrishnan, R.: Mondrian Multidimensional K-Anonymity. In: Proceedings of the 22nd International Conference in Data Engineering, p. 25. (2006)
3. Samarati, P.: Protecting Respondents' Identities in Microdata Release. IEEE Transactions on Knowledge and Data Engineering 13(6), 1010–1027 (2001)
4. Gruteser, M., Grunwald, D.: Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In: Proceedings of the 1st International Conference on Mobile Systems, Applications, and Services, pp. 31–42. (2003)
5. Gedik, B., Liu, L.: Protecting Location Privacy with Personalized k-Anonymity: Architecture and Algorithms. IEEE Transactions on Mobile Computing 7(1), 1–18 (2008)

6. Kalnis, P., Ghinita, G., Mouratidis, K., Papadias, D.: Preventing Location-Based Identity Inference in Anonymous Spatial Queries. IEEE Transactions on Knowledge and Data Engineering 19(12), 1719–1733 (2007)
7. Machanavajjhala, A., Gehrke, J., Kifer, D., Venkitasubramaniam, M.: ℓ–Diversity: Privacy Beyond k–Anonymity. In: Proceedings of the 22nd International Conference on Data Engineering, p. 24. (2006)
8. Li, N., Li, T., Venkatasubramanian, S.: t–Closeness: Privacy Beyond k–Anonymity and ℓ–Diversity. In: Proceedings of the 23rd International Conference on Data Engineering, pp. 106–115. (2007)
9. Wong, R.C., Fu, A.W., Wang, K., Pei, J.: Minimality Attack in Privacy Preserving Data Publishing. In: Proceedings of the 33rd International Conference on Very Large Data Bases, pp. 543–554. (2007)
10. Dwork, C.: Differential Privacy. Automata, Languages and Programming 4052, 1–12 (2006)
11. Bamba, B., Liu, L., Pesti, P., Wang, T.: Supporting Anonymous Location Queries in Mobile Environments with Privacy Grid. In: Proceedings of the 17th International World Wide Web Conference, pp. 237–246. (2008)
12. Xue, M., Kalnis, P., Pung, H.K.: Location Diversity: Enhanced Privacy Protection in Location Based Services. In: Proceedings of the 4th International Symposium on Location and Context Awareness, pp. 70–87. (2009)
13. Ghinita, G., Zhao, K., Papadias, D., Kalnis, P.: A Reciprocal Framework for Spatial k-Anonymity. Journal of Information Systems 35(3), 299–314 (2010)
14. Khoshgozaran, A., Shahabi, C.: Blind Evaluation of Nearest Neighbor Queries Using Space Transformation to Preserve Location Privacy. In: Proceedings of the 10th International Conference on Advances in Spatial and Temporal Databases, pp. 239–257. (2007)
15. Liu, X., Schrack, G.: Encoding and Decoding the Hilbert Order. Software-Practice and Experience 26(12), 1335–1346 (1996)
16. Kido, H., Yanagisawa, Y., Satoh, T.: An Anonymous Communication Technique Using Dummies for Location-Based Services. In: Proceedings of the IEEE International Conference on Pervasive Services, pp. 88–97. (2005)
17. Duckham, M., Kulik, L.: A Formal Model of Obfuscation and Negotiation for Location Privacy. In: Proceedings of the 3rd International Conference on Pervasive Computing, pp. 152–170. (2005)
18. Cheng, R., Zhang, Y., Bertino, E., Prabhakar, S.: Preserving User Location Privacy in Mobile Data Management Infrastructures. In: Proceedings of the 6th Workshop on Privacy Enhancing Technologies, pp. 393–412. (2006)
19. Yiu, M.L., Jensen, C.S., Huang, X., Lu, H.: SpaceTwist: Managing the Trade-Offs Among Location Privacy, Query Performance, and Query Accuracy in Mobile Services. In: Proceedings of the 24th International Conference on Data Engineering, pp. 366–375. (2008)
20. Ghinita, G., Kalnis, P., Skiadopoulos, S.: PRIVE: Anonymous Location-Based Queries in Distributed Mobile Systems. In: Proceedings of the 16th International Conference on World Wide Web, pp. 371–380. (2007)
21. Mokbel, M.F., Chow, C., Aref, W.G.: The New Casper: Query Processing for Location Services Without Compromising Privacy. In: Proceedings of the 32nd International Conference on Very Large Data Bases, pp. 763–774. (2006)
22. Lee, K.C.K., Lee, W.C., Leong, H.V., Zheng, B.: OPAQUE: Protecting Path Privacy in Directions Search. In: Proceedings of the 25th International Conference on Data Engineering, pp. 1271–1274. (2009)

23. Xu, T., Cai, Y.: Feeling-Based Location Privacy Protection for Location-Based Services. In: Proceedings of the 16th ACM Conference on Computer and Communications Security, pp. 348–357. (2009)
24. Soriano, M., Qing, S., Lopez, J.: Time Warp: How Time Affects Privacy in LBSs. In: Proceedings of the 12th International Conference on Information and Communications Security, pp. 325–339. (2010)
25. Shokri, R., Theodorakopoulos, G., Boudec, J.Y.L., Hubaux, J.P.: Quantifying Location Privacy. In: Proceedings of the 32nd IEEE Symposium on Security and Privacy, pp. 247–262. (2011)
26. Agrawal, R., Kiernan, J., Srikant, R., Xu, Y.: Order Preserving Encryption for Numeric Data. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 563–574. (2004)
27. Wong, W.K., Cheung, D.W., Kao, B., Mamouslis, N.: Secure kNN Computation on Encrypted Databases. In: Proceedings of the 35th SIGMOD International Conference on Management of Data, pp. 139–152. (2009)
28. Khoshgozaran, A., Shahabi, C., Shirani-Mehr, H.: Location Privacy: Going beyond k-Anonymity, Cloaking and Anonymizers. Journal of Knowledge and Information Systems 26(3), 435–465 (2011)
29. Papadopoulos, S., Bakiras, S., Papadias, D.: Nearest Neighbor Search with Strong Location Privacy. VLDB Endowment 3(1-2), 619–629 (2010)