Feeling-based Location Privacy Protection for Location-based Services

Toby Xu
Department of Computer Science
Iowa State University
Ames, Iowa 50011
U.S.A.
tobyxu@cs.iastate.edu

Ying Cai
Department of Computer Science
Iowa State University
Ames, Iowa 50011
U.S.A.
yingcai@cs.iastate.edu

ABSTRACT

Anonymous location information may be correlated with restricted spaces such as home and office for subject re-identification. This makes it a great challenge to provide location privacy protection for users of location-based services. Existing work adopts traditional K-anonymity model and ensures that each location disclosed in service requests is a spatial region that has been visited by at least K users. This strategy requires a user to specify an appropriate value of K in order to achieve a desired level of privacy protection. This is problematic because privacy is about feeling, and it is awkward for one to scale her feeling using a number. In this paper, we propose a feeling-based privacy model. The model allows a user to express her privacy requirement by specifying a *public* region, which the user would feel comfortable if the region is reported as her location. The popularity of the public region, measured using entropy based on its visitors' footprints inside it, is then used as the user's desired level of privacy protection. With this model in place, we present a novel technique that allows a user's location information to be reported as accurate as possible while providing her sufficient location privacy protection. The new technique supports trajectory cloaking and can be used in application scenarios where a user needs to make frequent location updates along a trajectory that cannot be predicted. In addition to evaluating the effectiveness of the proposed technique under various conditions through simulation, we have also implemented an experimental system for location privacy-aware uses of location-based services.

Categories and Subject Descriptors

H.2.0 [**Database Management**]: General—*Security, integrity, and protection*; H.2.8 [**Database Management**]: Database Applications —*Spatial databases and GIS*

General Terms

Security, Algorithm, Performance, Experimentation

CCS'09, November 9–13, 2009, Chicago, Illinois, USA.

Keywords

K-anonymity model, feeling-based privacy model, location privacy, trajectory cloaking, location-based services

1. INTRODUCTION

While location-based services (LBSs) offer significant opportunities for a broad range of markets, they present users significant privacy threats. An obvious one is service anonymity threat, i.e., the potential exposure of service uses. Just like regular Internet access, a user may not want to be identified as the subscriber of some LBS, especially when the service is sensitive. Another threat, which is more serious, is location privacy. A user's location disclosed in her service request may reveal sensitive private information such as health conditions, lifestyles, and so on. In particular, it has the potential to allow an adversary to locate the subject and result in physical harm.

For self-protection, it is natural and necessary for a user to withhold her true identity when requesting an LBS. However, simply using a pseudonym, or not using any identifier at all, is not sufficient. This is due to the fact that a user's location itself may be correlated with restricted spaces such as house and office to reveal her real-world identity. For example, if a location belongs to a private property, then the adversary can derive that the user is most likely the owner of the property. A single location sample may not be linked directly to a particular user, but the accumulation of a time-series sequence of her location samples will eventually reveal her identity [16, 29]. Once the user is identified, all her visits may be disclosed.

The above problem, known as *restricted space identification*, has motivated a series of research effort on location depersonalization. The proposed techniques reduce location resolution and can be classified into two categories according to their purposes:

- **Anonymous service uses**: The techniques in this category (e.g., [15], [13], [25], [7], [9]) aim at preventing location information collected by the LBS providers from being used to identify a service user. When a user requests an LBS, these schemes compute a cloaking box that contains the user and at least $K - 1$ neighbors. This box is then reported as the user's location to the corresponding LBS provider. Since each of the K users may be the one who requests the service, this strategy provides a certain level of guarantee that the service user cannot be identified.

- **Location privacy protection**: The above techniques protect users' anonymity in service uses, but not their location privacy. By correlating with restricted spaces, an adversary may be able to identify the users in a cloaking box. The adversary may not know who requests the service, but knows the location of these users – they are all in the area at the time of the service request. To address this problem, the technique proposed in [30] ensures that each location reported on behalf of a user is a cloaking region that contains at least K different *footprints*, each being a location sample collected at some time point. A spatial region with K different footprints means it has been visited by K different people. An adversary may manage to identify these visitors, but will not know who was there at time of the service request.

In this paper, we investigate location depersonalization from the perspective of location privacy protection. To our knowledge, exploring historical location samples for location depersonalization is the only practical solution up to date that can prevent anonymous location information from being correlated with restricted spaces to derive who's where at what time. This strategy can be applied to depersonalize not only individual location samples, but also location samples that form a user's trajectory. Nevertheless, the technique proposed in [30] has several problems. Like all other techniques, it adopts the traditional K-anonymity model [35, 36]. To request a desired level of privacy protection, a user needs to specify the value of K. Unfortunately, choosing an appropriate K value can be difficult. For example, why would a user feel that her privacy is well-protected if $K = 20$, but not if $K = 19$? Ultimately, privacy is about feeling, and it is awkward for one to scale her feeling using a number. As in the above example, it is hard to tell the difference between the two K values in terms of privacy feeling. A user can always choose a large K to ensure a sufficient privacy protection, but this will result in unnecessary reduction of location resolution. A very coarse location will make it difficult to provide a meaningful LBS.

In addition to privacy modeling, robustness is another issue. Ensuring each reported location has been visited by at least K different users may not provide privacy protection at the level of K. Indeed, it can achieve so only when these K users have an equal chance of visiting the region, i.e., they leave the same amount of footprints in the area. In reality, a spatial region may be visited by many people, but some may have a dominant presence (e.g., in an office). In this case, the one who is known to be dominant is most likely to be the subject. Furthermore, for continuous LBSs, where users needs to make frequent location updates, the proposed technique requires a user to report her trajectory ahead of her travel in order to compute a cloaking trajectory that has been traversed earlier by at least $K - 1$ other users. This requirement prevents the technique from being used when a user's movement is not pre-determined.

This paper addresses the above problems. We propose a feeling-based privacy model for location privacy protection. Our idea is to let a user express her privacy requirement by specifying a *public region*, instead of a value of K. A spatial region is considered a user's public region if the user feels comfortable that the region is reported as her current location when the user is inside the region. For example, a shopping mall can be a user's public region, if the user does not mind that the mall is disclosed as her location when she requests an LBS in it. Given a public region specified by a user, we apply the concept of *entropy* to measure its popularity based on the footprints collected from the visitors of the region. This popularity is then used as the user's privacy requirement: For each location disclosed on behalf of the user, we ensure that the popularity of this location is no less than that of the specified public region. With this privacy model in place, we investigate the challenges of location depersonalization in the context of continuous LBSs. We present a novel technique that allows a user's time-series sequence of location information to be reported as accurately as possible while ensuring that her location privacy requirement is always met. Unlike the existing approach [30], the new technique cloaks a user's movement on the fly without having to know the moving trajectory in advance. As such, it can be used in application scenarios where a user needs to make frequent location updates along a trajectory that is not predetermined. We evaluate the performance of the proposed technique under various conditions through simulation. Moreover, we have implemented a prototype that supports location privacy-aware uses of LBSs.

The rest of this paper is organized as follows. In Section 2, we present our feeling-based privacy model and introduce some definitions. In Section 3, we propose an algorithm for on-the-fly trajectory cloaking. We evaluate the performance of the proposed technique through simulation in Section 4, and present a prototype that we implement in Section 5. We discuss related work in more detail in Section 6, and conclude this paper in Section 7.

2. FEELING-BASED PRIVACY MODEL

An anonymous location disclosed for an LBS may be correlated with restricted spaces to identify a set of possible service requestors. The more popular a spatial region is, the more difficult it is for an adversary to single out the true requestor. A user can express her desired level of protection by specifying a value of K: a spatial region disclosed on her behalf must have at least K different visitors. Alternatively, a user can specify a *public region* and request that her disclosed location must be at least as popular as that space. An example of public region can be some shopping mall in town. As compared to choosing a number of K, it is much more intuitive for a user to express her privacy requirement by identifying a spatial region which she feels comfortable is reported as her location should she request an LBS from it. We refer to this approach as feeling-based privacy modeling.

When a location is disclosed for an LBS on a user's behalf, it must be at least the same popular as the public region she specifies. The problem now is how to measure the popularity of a spatial region. The number of its visitor along is not sufficient to quantify its popularity, because some people may have a dominant presence in that space. If an LBS is requested from an office, then the office staff is more likely to be the service requestor, even if the office has many visitors. To address this problem, we borrow the concept of *entropy* from Shannon's information theory [28]. Suppose we can collect location samples from cellular phone users. These location samples, each called a *footprint*, can then be used to measure the popularity of a spatial region as follows.

DEFINITION 1. *Let R be a spatial region and $S(R) = \{u_1, u_2, \cdots, u_m\}$ be the set of users who have footprints in R. Let n_i $(1 \leq i \leq m)$ be the number of footprints that user u_i has in R, and $N = \sum_{i=1}^{m} n_i$. We define the* entropy *of R as $E(R) = -\sum_{i=1}^{m} \frac{n_i}{N} \log \frac{n_i}{N}$, and the* popularity *of R as $P(R) = 2^{E(R)}$.*

The value of $E(R)$ can be interpreted as the amount of additional information needed for the adversary to identify the service user from $S(R)$ when R is reported as her location in requesting an LBS. According to the above definition, we have $1 < P(R) \leq m$. $P(R)$ has the maximum value m when every user in $S(R)$ has the same number of footprints in R. On the other hand, $P(R)$ has the minimum value when one user in $S(R)$ has $N - m + 1$ footprints in R while each of the rest has only 1. We have the following two observations. First, $P(R)$ is higher if m is larger. In other words, a region is more popular if it has more visitors. Second, $P(R)$ has a lower value if the distribution of footprints is more skewed. If some users are dominant in the region, $P(R)$ will be much less than m. In this case, R needs to be enlarged to contain more users in order to have a required popularity.

Let R be a user's public region. When the user requests a sporadic LBS, where the request can be seen as an independent event, we can find a cloaking box that 1) contains the user's current position, 2) has a popularity that is no less than $P(R)$, and 3) is as small as possible, and then report this box as the user's location. When the user requests a continuous LBS, a time-series sequence of cloaking boxes will be reported that form a trajectory. In this case, simply ensuring that each cloaking box has a popularity no less than $P(R)$ does not protect the user's location privacy at her desired level. This is due to the fact that the adversary can narrow down the list of possible service users by finding the common visitors of these cloaking boxes. To prevent such attack, we must use the footprints of the common set of users, instead of all visitors of the regions, in computing the popularity of each cloaking box. We define the popularity of a spatial region with respect to a given set of users as follows.

DEFINITION 2. *Given a spatial region R, and a user set $U = \{u_1, u_2, \cdots, u_{m'}\} \subseteq S(R)$, the* entropy *of R with respect to U is $E_U(R) = -\sum_{i=1}^{m'} \frac{n_i}{N'} \log \frac{n_i}{N'}$, where n_i is the number of footprints that u_i has in R, and $N' = \sum_{i=1}^{m'} n_i$. The* popularity *of R with respect to U is $P_U(R) = 2^{E_U(R)}$.*

When a sequence of cloaking boxes are generated on a user's behalf, we must ensure that the popularity of each cloaking box with respect to the common set of visitors is no less than that of the user's public region. In other words, the trajectory formed by these cloaking boxes must be a *P-Popular Trajectory* (PPT), which is formally defined below:

DEFINITION 3. *Let $T = \{R_1, R_2, \cdots, R_n\}$ be a sequence of cloaking boxes generated for a user, and $S(R_i)$ ($1 \le i \le n$) the set of people who have footprints in R_i. We say T is the user's PPT if for each R_i, it satisfies that (1) R_i covers the user's position at the time when R_i is disclosed, and (2) $P_S(R_i) \ge P(R)$, where $S = \bigcap_{1 \le i \le n} S(R_i)$ and R is the public region specified by the user.*

Given a trajectory $T = \{R_1, R_2, \cdots, R_n\}$, we define its resolution to be $|T| = \frac{\sum_{i=1}^{n} Area(R_i)}{n}$, where $Area(R_i)$ denotes the area of box R_i. For location privacy protection, a trajectory formed by the location samples disclosed on a user's behalf must be a PPT. Meanwhile, its resolution needs to be as fine as possible to guarantee the quality of the required LBS services. In the rest of this paper, we focus on how to generate such a PPT for a user to entertain a continuous LBS.

3. TRAJECTORY CLOAKING

Similar to [30], we assume mobile clients communicate with LBS providers through a trusted central location depersonalization server (LDS) managed by the clients' cellular service carriers. For LBSs that require user authentication (e.g., for service charges), we assume anonymous authentication (e.g., [18], [27], [21]) is used. The carriers offer the depersonalization services as a value-added feature to their clients, and supply the LDS with an initial footprint database that contains location samples collected from their clients (e.g., through regular phone calls). These location samples will be used to compute the popularity of a spatial region and for trajectory cloaking. The database will be expanded with the location data obtained from mobile users in their requests of LBSs.

We assume each client configures her privacy requirement by specifying a public region. When a user requests an LBS, she also informs the LBDs a travel bound B, a rectangular spatial region that bounds her travel during the service session. In response, the LDS randomly generates a service session ID and contacts the service provider. After establishing a service session, the service user periodically reports her current location to the LDS. For each location update, the LDS computes a cloaking box which contains the service user's current location, and exports this box along with the session ID to the corresponding LBS provider. The information received from the provider is then forwarded back to the service user. As mentioned early, to prevent restricted space identification, the trajectory created by the sequence of cloaking boxes must be a PPT that satisfies the user's privacy requirement. The key issue is how to find a common set of users for cloaking so that the trajectory, which is undetermined, can have a resolution that is as fine as possible. In the following subsections, we first describe the main data structure used for indexing the location samples stored in the footprint database, and then present a heuristic algorithm for trajectory cloaking.

3.1 Data Structure

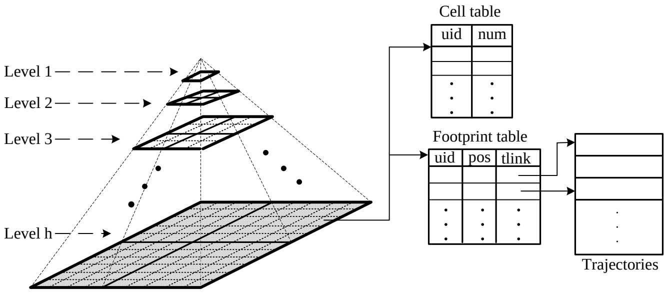


Figure 1: Data structure

We partition the network domain recursively into cells in a quad-tree style. The partitioning stops when the size of cells becomes less than a threshold (our implementation sets each cell to be at least 100×100 $meter^2$). All the cells generated in the partitioning form a pyramid structure as shown in Figure 1. Suppose the partitioning stops at the h^{th} recursion, then the pyramid has a height of h. The top level in the pyramid is level 1 and has only one grid cell that covers the whole network domain. Each grid cell except the ones at the bottom level is composed of four cells at the next lower level, which we refer to as its child cells.

Each cell at the bottom level h keeps a footprint table and a user table. The footprint table stores the footprints the cell contains, and each tuple of the table is a record of $(uid, pos, tlink)$, where uid is the identity of the mobile user that a footprint belongs to, pos is the coordinates of a footprint, and $tlink$ is a pointer that links to the corresponding trajectory stored in the database. The user table records the number of footprints a user has in the cell, and each tuple of the table is a record of (uid, num), where num is the number of footprints the user has in the cell. For each cell not at the bottom level, we also keeps a user table, which is derived from the user tables corresponding to its four child cells.

3.2 Generating PPT

We now discuss how to generate a PPT for a service user. Given the user's public region R, the LDS computes its popularity $P(R)$ using the cells at the bottom level that overlap with

R. When the user makes the first location update, the server selects a set of users, which we will refer to as a *cloaking set*. The footprints of this set of users are then used for location cloaking whenever the service user makes a location update.

3.2.1 Selecting cloaking set

It may first appear that we can determine the cloaking set, denoted as S, by finding the set of users who have footprints closest to the starting point of the service user. This simple solution minimizes the size of the first cloaking box. However, as the service user moves, the users in S may not have footprints that are close to her current position. As a result, the size of the cloaking boxes may become larger and larger, making it difficult to guarantee the quality of LBS. Thus, when selecting the cloaking set, we should consider its affect on the cloaking of not only the user's first but all location updates in the LBS. But the challenge is that the service user's route is not predetermined, and thus the LDS cannot figure out whose footprints will be closer to the service user during her travel. To address this challenge, our idea is to find those users who have visited most places in the service user's travel bound B and use them to create the cloaking set. As these users have footprints spanning the entire region B, it will help generate a PPT with a fine resolution.

We say a user is *l-popular within B*, if she has footprints in every cell at level l that overlaps with B. According to the pyramid structure, cells at level with a larger l have a finer granularity. This implies that given an l-popular user, the larger the value of l is, the more popular the user is. Figure 2 shows an example in which a network domain is partitioned into a 4-level pyramid (There are 1, 4, 16, 64 cells at each level respectively from top to bottom). It also shows a travel bound B and the footprints inside it. The footprints in different colors belong to different users. u_1, u_2, and u_3 are three 2-popular users within B because they have footprints in the two cells at level 2 of the pyramid which overlap with B; u_2, u_3 are two 3-popular users within B since they have footprints in all four cells at level 3 that overlap with B; only u_3 is 4-popular since she is the only one who has footprints in all the sixteen cells at level 4 that overlap with B.

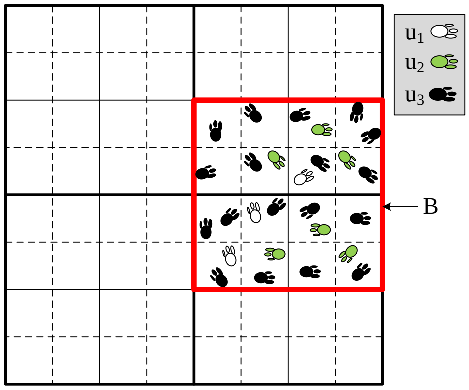


Figure 2: A travel bound and footprints inside

Based on the above definitions, we now present a simple but effective algorithm that can find a cloaking set for trajectory cloaking. The pseudo code is given in Algorithm 1. In this algorithm, the LDS sorts the users in $S(B)$ according to their popularity at level l, and selects the most popular users in $S(B)$ as the cloaking set, starting from the bottom to top of the pyramid. Let C_l denote the set of cells at level l in the pyramid, C'_l the set of cells in C_l that overlap with B, and S_l the set of users who are l-popular within B. The LDS first finds S_h. Since level h is the bottom level, these users are the most popular users in $S(B)$. To find S_h (i.e., the users who have visited all the cells in C'_h), the LDS simply joins the user tables of these cells on column uid (line 6-7). Next, the LDS computes the popularity of B with respect to S_h using their footprints in B. If the popularity $P_{S_h}(B)$ is less than $P(R)$, it means that cloaking with the footprints of the users in S_h cannot provide the desired level of privacy protection for the service user. In this case, the LDS considers the cells one level higher, i.e., level $h-1$ (line 9), and computes S_{h-1} and $P_{S_{h-1}}(B)$ similarly. This procedure is repeated until at some level l the popularity $P_{S_l}(B)$ is no less than $P(R)$. The complexity of this algorithm is determined by the cost of computing user set S_l at each level from bottom to top. Let m denote the number of users in $S(B)$ and k the number of cells in C'_h. Then, the cost of joining two user tables is $O(m)$, and the cost of joining user tables at bottom level (i.e., computing S_h) is $O(k \cdot m)$. According to the pyramid structure, the number of cells at a certain level that overlap with B is about one fourth of those at the next lower level. Thus, the total cost of finding S_l on all levels is $O(k \cdot m)$.

Algorithm 1 SelectCloakingSet($P(R)$, B)

```
 1: $U \leftarrow \emptyset$ {$U$ keeps the cloaking set}
 2: $l \leftarrow h$
 3: while $U \subset S(B)$ and $P_U(B) < P(R)$ do
 4:     {Get cells at level $l$ overlapping with $B$}
 5:     $C'_l \leftarrow Overlap(C_l, B)$
 6:     {Join user tables of $C'_l$ by column $uid$}
 7:     $T \leftarrow Join(C'_l, uid)$
 8:     $U \leftarrow S_l \leftarrow T.uid$
 9:     $l \leftarrow l - 1$
10: end while
11: return $U$
```

The above algorithm checks the users level by level, from the bottom to top. If a user is l-popular within B, it must also be $(l-1)$-popular within B. Thus, each time the algorithm checks the cells at a higher level, the cloaking set is expanded to include more users. As long as $P(R) \leq P(B)$ (i.e., a user's public region is at most the same popular as that of her travel bound), the algorithm will find a sufficient number of visitors within B for the cloaking set. In the worst case, all users in $S(B)$ are included in the cloaking set. On the other hand, if $P(B) < P(R)$, the LDS does not need to find a cloaking set. It can simply compute a spatial region that contains B and has a popularity no less than $P(R)$, and always report this region as the user's location as long as it moves inside B.

3.2.2 Computing cloaking boxes

During a service session, the service user updates a time-series sequence of locations. For each location update p, the LDS computes a cloaking box b using the footprints of users in the cloaking set U. We develop a heuristic algorithm which computes the cloaking box b as small as possible, and ensures that $P_U(b) \geq P(R)$. The pseudo code is given in Algorithm 2.

Given a location update p, the LDS first initializes the cloaking box b to p which is the smallest cloaking box only containing the service user herself. The LDS also initializes a searching box b' to the cell that contains p at level l where the cloaking set U is selected in Algorithm 1, since it contains footprints of all users in the cloaking set. Then, for each user in U, the LDS gets the set of her footprints F_u which are inside b' but outside b, and in F_u the LDS finds the closest one to p (line 7-8). Next, the LDS collects these footprints in set F, and computes the cloaking box b as the minimal bounding box (MBB) of the footprints in F (line 11). If b already contains all footprints of U in b', the LDS expands the searching box b' by merging itself with its adjacent cells at the bottom level (line 13-16). The above procedure is repeated until $P_U(b) \geq P(R)$, and the resulting cloaking box b is reported as the service user's location to the external service provider.

Algorithm 2 Cloak(p, $P(R)$, U)

```
 1: $F \leftarrow \emptyset$
 2: $l \leftarrow$ the level where $U$ is determined
 3: $b \leftarrow p$
 4: $b' \leftarrow$ the cell in $C_l$ that contains $p$
 5: while $P_U(b) < P(R)$ do
 6:     for all $u \in U$ do
 7:         $F_u \leftarrow$ the footprints of $u$ in $b' - b$
 8:         $f_u \leftarrow$ the closest footprint to $p$ in $F_u$
 9:         $F \leftarrow F + \{f_u\}$
10:     end for
11:     $b \leftarrow MBB(F)$
12:     if $b$ contains all footprints of $U$ in $b'$ then
13:         {get cells at bottom level adjacent to $b'$}
14:         $C' \leftarrow Adjacent(b', h)$
15:         {merging the cells in $C'$ with $b'$}
16:         $b' \leftarrow b' \bigcup C'$
17:     end if
18: end while
19: return $b$
```

4. PERFORMANCE STUDY

In this section, we evaluate the effectiveness of the proposed technique under various conditions using location data synthetically generated based on a real road map. For comparison purpose, we have implemented two other approaches. The first one, which we will refer to as *Naive*, assumes the location updates made a service user are independent to each other. For each location update, Naive just finds a cloaking box which satisfies the three conditions as described at the beginning of Section 3, and reports it as the service user's location in her service request. Note that this scheme may not protect a user's location privacy at her desired level when she makes a time-series sequence of location updates. The second approach is referred to as *Plain* hereafter. This scheme determines the cloaking set for the service users by finding the footprints closest to her start position. After fixing the cloaking set, Algorithm 2 is applied to compute the cloaking boxes for the service user during her entire service session. To ease our presentation, we will refer to our proposed technique as *Advanced*.

We modify the simulator *Network-based Generator of Moving Objects* [5] to generate mobile nodes and simulate their movement on the real road map of Oldenburg, Germany, a city about 15×15 km^2. We extract four types of roads from the road map, primary road (interstate expressway), secondary road (state road), connecting road and neighborhood road as defined in census TIGER/Line [2]. In our simulation, mobile nodes change their speeds at each intersection, and the moving speed on a road follows a normal distribution determined by the road type. The mean speeds and the standard deviations of moving speeds on all road types are listed in Table 1. We generate a footprint database that contains a certain number of trajectories, which are assigned to 2000 users. The number of trajectories each user has follows a normal distribution with a standard deviation 0.1. These trajectories are indexed using the grid-based approach discussed in the Section 3.1. For each simulation, we generate a set of LBS requests. Each service request contains a user's ID, a public region, and a travel bound. The start position is randomly selected within the travel bound, and the service user moves randomly within the travel bound, i.e., when arriving at an intersection, she randomly chooses a direction to move on. We assume a user's travel distance is proportional to the size of the travel bound, and she makes a location update every 100 meters she moves. Other parameters used in our study are given in Table 2. Unless otherwise specified, the default values are used.

In our study, we are mainly interested in the following two performance metrics. One is *cloaking area*, defined to be the average area of cloaking boxes in a cloaking trajectory. The other one is *protection level*. Given a cloaking trajectory, we measure its protection level using the ratio between the average popularity of its cloaking boxes with respect to the common set of users who have visited all of them and the popularity of the user specified public region. Clearly, the protection level must be at least 1, otherwise the cloaking trajectory fails to protect the service user's location privacy at the required level. In the following subsections, we report how the performance of the three techniques is affected by various factors.

Table 1: Traffic parameters

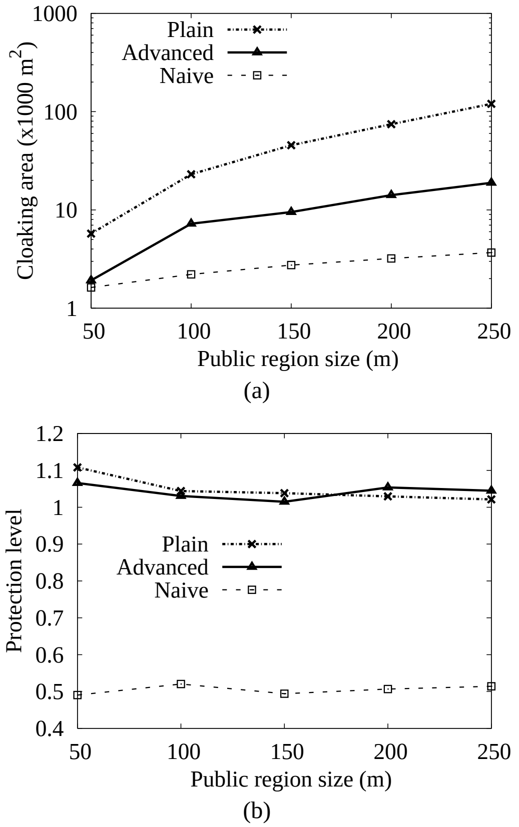
Road type	Mean speed	Standard deviation
Primary	$100km/h$	$20km/h$
Secondary	$60km/h$	$15km/h$
Connecting	$45km/h$	$10km/h$
Neighborhood	$30km/h$	$5km/h$

Table 2: Experiment settings

parameter	range	default	unit
Number of users	2000	2000	$unit$
Public region size	50 - 250	150	$meter$
Trajectory database size	$100K - 300K$	$200K$	$unit$
Travel bound size	$2 - 6$	4	km
Travel distance	$2 - 6$	4	km
Service request number	300	300	$unit$
Minimum cell size	100×100	100×100	$meter^2$

4.1 Effect of Privacy Requirement

This study investigates the impact of privacy requirement (i.e., the popularity of the public region specified by a service user) on the performance of the three techniques. We generated 300 service requests. Each request has a travel bound of a 4×4 km^2 square region, and the travel distance of the corresponding user during her service session is 4 km. Each service user specifies her public region as a square region which contains her start position. The size of a public region, measured by the side length of the square, is varied from 50 to 250 meters. The performance results are plotted in Figure 3. Figure 3(a) shows that when the size of the public region increases, the average cloaking area under all the three schemes increases. This is due to the fact that a larger public region is likely to contain more people's footprints and have a higher popularity. To satisfy a higher level of privacy requirement, a cloaking box needs to be larger to include more people. This study also shows that Plain always has a much larger cloaking area as compared to the other two approaches. This scheme does does not take user popularity into consideration when selecting a user's cloaking set. When some unpopular users are selected in a cloaking set, the cloaking boxes generated for the future movement of a service user will become larger and larger in order to contain all users in the cloaking set. On the other hand, Naive has the smallest cloaking area. This scheme does not consider the correlation of the cloaking boxes in a trajectory, just cloaking each location with a bounding box that is as small as possible and has a popularity no less than that of the public region. The problem is, simply ensuring that each cloaking box satisfies the privacy requirement does not protect a user's privacy at her specified level. This is confirmed in Figure 3(b). It shows that the protection level of Naive is constantly lower than 1. As for Plain and Advanced, they both guarantee that the actual protection level is no less than required.



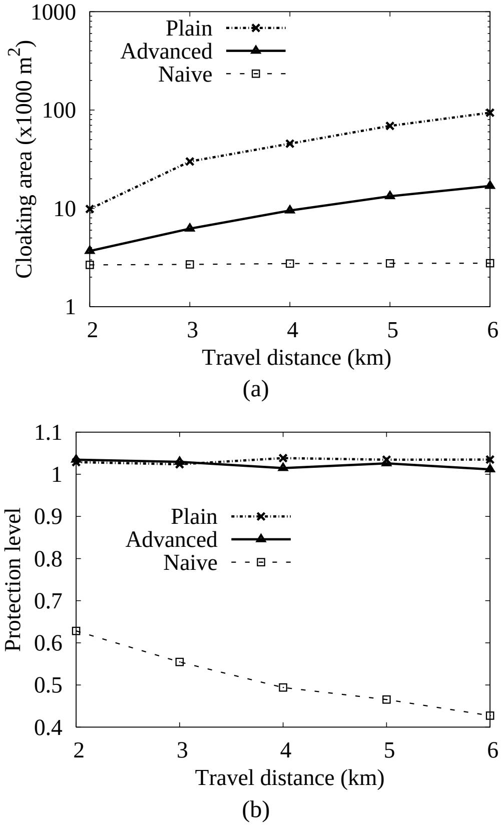
(a)

(b)

Figure 3: Effect of privacy requirement

4.2 Effect of Travel Distance

In this study, we investigated the impact of travel distance on the performance of the three techniques. In each simulation run, we set the public region as a 150×150 m^2 square, and generated 300 service requests. The travel distance is varied from 2 km to 6 km, and accordingly the side length of travel bound is varied from 2 km to 6 km. The performance results are shown in Figure 4(a) and (b). Figure 4(a) shows that under both Plain and Advanced, the average cloaking area increases as the travel distance increases. However, Plain performs much worse than Advanced. The reason behind is explained as follows. When the travel distance is larger, the trajectory of the service user tends to traverse through a larger region. With an unpopular user in a cloaking set, it is more difficult to find their footprints close for each location update in the trajectory. Plain performs worse because in average it includes more unpopular users in a cloaking set. On the other hand, the cloaking area under Naive remains almost constant as the travel distance changes. It is due to the fact that Naive assumes each location update is an independent event. For each location update, it simply finds the nearest footprints to cloak. As such, the cloaking area is irrelevant to the number of location updates in the trajectory. Again, this approach cannot be used for location privacy protection when a user has to report her location periodically in a service session. Figure 4(b) shows the protection level of Naive decreases as the travel distance increases. Since each location update is cloaked independently in Naive, a longer trajectory tends to have a less number of users who have visited all cloaking boxes in the trajectory, and thus has a lower popularity with respect to this common set of users. In contrast, the privacy level of neither Plain nor Advanced is much affected by the variance of travel distance.



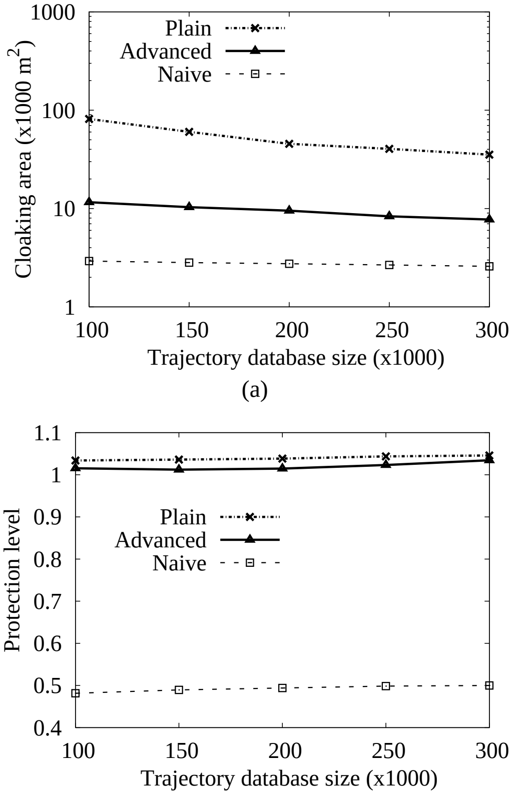
(a)

(b)

Figure 4: Effect of travel distance

4.3 Effect of Footprint Database Size

This study investigates the impact of the number of trajectories in the footprint database on the performance. We varied the number of trajectories in the database from 100,000 to 300,000. The performance results are plotted in Figure 5(a) and (b). It is shown in Figure 5(a) that all schemes have better cloaking results when the database contains more trajectories. Clearly, more historical trajectories mean that more footprints collected in a fixed spatial region. As a result, a smaller cloaking box may be populous enough to meet the privacy requirement. By adding a service user's moving route to the database for future cloaking, our technique can generate better cloaking results. This feature makes it especially attractive for large-scale LBS that consists of a large number of users. Figure 5(b) again shows that the protection level of Naive is constantly lower than 1. On the other hand, the protection level of both Plan and Advanced is always above 1.



(a)

(b)

Figure 5: Effect of database size

5. IMPLEMENTATION

We have implemented an experimental system based on the technique presented in the previous sections. The prototype, called *location privacy aware gateway* (LPAG), has two software components, client and server. Client is implemented in C# using .Net Compact Framework 1.0. It runs on Windows Mobile 2003 platform and we have tested it with two types of mobile devices, HP IPAQ 6515 and HP IPAQ 4310 as shown in Figure 6. The former is a smart phone with a built-in 4-channel GPS receiver. The device communicates with the server through AT&T's GPRS wireless data service. As long as it is within the region covered by the carrier's service network, it can stay connected to the server which is located in our lab. The other type of client device, namely HP IPAQ 4310, is a regular pocket PC which connects with the server through our university's campus wireless network, which limits its roaming area to be within our campus. To make it position-aware, we bundle it with an external 16-channel GPS receiver, which provides position information through bluetooth connection. The server component is implemented in C# using .Net Framework 1.0. It manages the historical location data and corresponding indices using MySQL 5.0, and cloaks mobile clients' location updates using the proposed techniques when they request LBSs. In a separate research project, we have a implemented a location-based service system called *ePostit* [6]. This system allows one to publish a geo-referenced message, each associated with a geographic region. A message is delivered to a user when the user arrive at the corresponding region. In our experiment, we also plant a number of spatial messages in our campus and let a user entertain the services provided by ePostit through the LPAG.

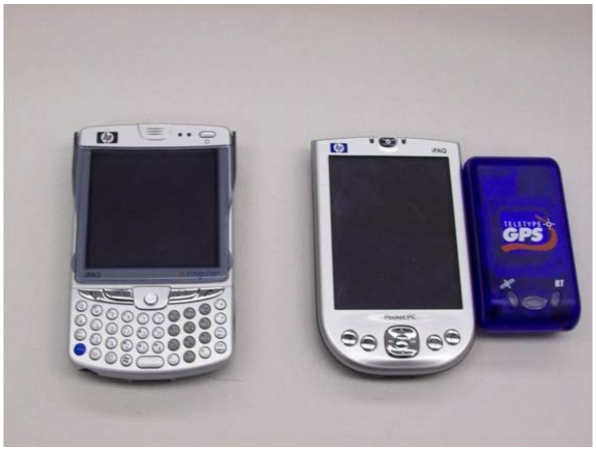


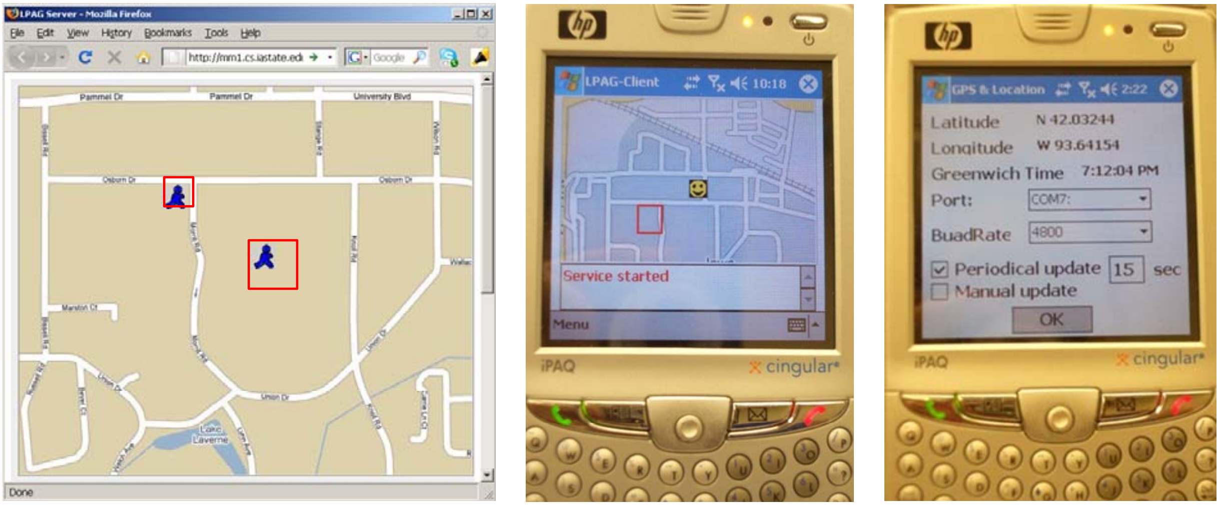
Figure 6: Client devices

Our test of LPAG consists of a location sampling phase, during which we collect users' footprints for location depersonalization. We create a number of client accounts, and carry the devices and have a walk around the campus, during which the devices makes periodical location update to the server. After a trajectory is collected, we randomly choose a client from the accounts created before, assign the trajectory to the client, and save it in the trajectory database in the server. In our testing of LPAG, we specify a rectangular region in the campus as the public region, and have a walk in the campus with a mobile device. During the walk, we send a sequence of queries to the server, each with our current position. For each query, the server generates a cloaking box using the proposed technique, and forwards it to ePostit. In response, the service provider delivers all the messages whose bounding boxes overlap with the cloaking box to the server, and the server forwards to the client only the ones whose bounding box contains the client's current position. In the following subsections, we introduce our system's user interfaces and discuss the experimental results collected in our field tests.

5.1 Server and Client User Interface

Figure 7 (a) shows the server interface. Every time the server receives a query from a client, it computes a cloaking box as the client's location in requesting the service. Then, the server displays the cloaking box and the client's position on the map. As the example shown in this figure, two clients and their cloaking boxes are displayed on the campus map.

When a mobile device is powered on, the client finds out the current position and then connects to the server. After initialization, the screen shows a local map as its background and marks the client's position by a small face icon (see Figure 7 (b)). At the beginning of a service session, the client can set the public region by clicking the touch screen to specify its top-left corner and bottom-right corner, and embed the public region in the query packet. In the example shown in Figure 7 (b), the client specifies the library as her public region which is marked by the red rectangle. In our experiments, the travel bound is set as the



Service started

Latitude N 42.03244
Longitude W 93.64154
Greenwich Time 7:12:04 PM
Port: COM7:
BuadRate 4800
☑ Periodical update 15 sec
□ Manual update
OK

(a) (b) (c)

Figure 7: Server and client interface

whole campus. Then, during the session, the client can choose to periodically update her location or manually update whenever she wants (see Figure 7 (c)).

5.2 Experimental Results

We first examine the system resources used by our code running on mobile devices.

CPU Utilization: We measure the CPU utilization of our client code on the smartphone using Xda pps [1], which allows one to monitor the CPU usage of all the processes running on a smart device. When the device is idling with no movement, the CPU utilization is about 1%, indicating that reading GPS position (every one second) does not take much computation. When the client moves around but does not make any location update, we observe that the CPU utilization is in between $4\% - 12\%$, as our code redraws the client's position on the map. When the client communicates with the server (e.g., location update, message delivery), the CPU utilization is in between $10\% - 25\%$.

Memory and Storage: Our client executable is only 120KB by itself. Since it is built on the .NET Compact Framework 1.0 and OPENNETCF 1.4, additional 2.5MB and 580KB files from the two platforms are needed, respectively. When running, our system has a memory footprint of 5.1MB, which is less than 10% of available main memory on HP IPAQ6515 (57.78MB) and HP IPAQ4310 (56.77MB). On both devices, our code can run simultaneously with other applications such as media player and Internet explorer.

We also examine two performance metrics which affect the usability of our system.

GPS Accuracy: Because of position deviation of the GPS receiver, the position reported to the server may be different from the actual position of a client. If the position deviation is large, the bounding box computed by the server may not contain the client's position, and the client may get the false query result (missing or downloading wrong messages). In our experiments, we have tested the accuracy of the two types of GPS in the campus area. The smartphone we use has a built-in 4-channel GPS, while the external GPS bundled with the pocket PC has 16 channels. To calculate the position, a GPS receiver needs to have signals from at least 4 satellites. In general, the more channels available, the more accurate position it can compute. Our tests show that the 16-channel GPS has 5 meters error in average and 8 meters error in maximum. While the 4-channel GPS performs worse. It has 7 meters error in average and 14 meters error in maximum. These tests indicate that in the worst case the server should expand the boundary of the cloaking box by 15 meters to ensure the cloaking box contains the client's actual position, and the bounding box of a message should not be smaller than $15m \times 15m$.

Response Time: The interval between the time a client sends a query and the time she receives the query result consists of four parts: (1) the time it takes to deliver the query from the client to the server, (2) the time the server uses to compute the cloaking box, (3) the time for the server to send the cloaking box to the service provider and receive candidate messages from the service provider, (4) the time it takes to download the resulting messages from the server to the client. Our experiments show that the server computes the cloaking box usually in less than 10 ms. In addition, the transmission speed between the server and the service provider is also very fast (>4MB/s) since they are connected with a high speed LAN. The bottleneck is the communication between the client and the server, i.e., part (1) and (4). The smartphone we use connects to our server via AT&T's GPRS, while the Pocket PC connects to our server via our campus's WLAN. In our test, we create a number of messages, some with simple text messages (1-5KB) and short audio clips (10-30KB), while the rest with video clips (100-300KB). Our tests show that for messages with simple text and audio clips, the smartphone and pocket PC can download them with a delay of less than 1 second and 3 seconds, respectively; for the messages with video clips, the pocket PC has a minimal delay of 5 seconds while the smartphone has a latency of more than 20 seconds. This study indicates that for cellular phones, our system is more appropriate for light-weight messages. Fortunately, this will not be a problem as the development of broadband wireless services provided by the cellular carriers.

6. RELATED WORK

In addition to [30], another work aimed at location privacy protection is [32]. The proposed technique lets a service user directly download location-based information from a service provider without having to report her location. It applies the theory of Private Information Retrieval (PIR) [34] to prevent an adversary from deriving the user's location according to the downloaded data. This scheme, however, requires a client to download about the square root of the total amount of data stored in the service provider. This presents a major burden to a mobile client since the database in the service provider may contain a large amount of location-based information. Some other related works are as follows.

Anonymous uses of LBSs: This problem was first investigated in [15]. The proposed solution reduces the accuracy of location information along spatial and/or temporal dimensions. When a client requests a service, the proposed scheme computes a cloaking box that contains the client and at least $K - 1$ others, and then uses this cloaking box as the client's location to request the service. If the resolution is too coarse for quality services, temporal cloaking is applied, i.e., delaying a user's service request. When more mobile nodes come near to the user, a smaller cloaking area can then be computed. This basic concept has since been improved by a series of work. The work in [13] considers allowing users to specify their own value of K and minimizing the size of cloaking boxes, a factor critical for the quality of location-based services. The techniques proposed in [25, 22, 7, 31] address the challenges of processing location-dependent queries with reduced location resolution. Preventing an adversary from identifying a subject based on her moving pattern was considered in [4] and [26]. The proposed techniques cloak a client's position using the neighbors that have been close to the client for some time period. All these techniques rely on a central anonymity server, which tracks the movement of mobile nodes and computes cloaking boxes upon requests. Anonymous usage of LBSs in fully distributed mobile peer-to-peer environments was investigated in [9] and [12]. Assuming mobile nodes trust each other, the proposed techniques let mobile nodes exchange location information and collaborate in computing cloaking boxes. More recent work [14, 8] assumes that users' actual positions are publicly known. To provide a service requestor K-anonymity protection, the proposed techniques ensure that her cloaking box should not only contain at least K users, but also be shared by at least K of these users. All the above techniques cloak a user's location with her current neighbors. As such, they can support only service anonymity, but not location privacy.

Trajectory perturbation: This problem was first investigated in [3], and the concept of *mix zone* was introduced to prevent revealing users' locations. A mix zone is defined to be a spatial region in which a mobile node does not report its location. When there are multiple nodes inside the same mix zone, they exchange their pseudonyms. After exiting the mix zone, these nodes start to use new pseudonyms in location updates, making it hard for an adversary to link incoming and outgoing paths of these nodes. While this approach relies on a set of pre-defined spatial regions for pseudonym exchange, the path confusion algorithm proposed in [19] allows mobile nodes to switch their pseudonyms when their paths are close to each other, say, within some threshold. Another strategy they proposed is to ensure that the time interval between two consecutive location reports is long enough so that each can be considered as an independent event [20]. These approaches reduce, but cannot prevent, location privacy risks. A partial trace, or just a single location sample, can be sufficient for an adversary to identify a user, thus knowing her whereabouts.

Privacy protection in opportunistic sensing and monitoring: The framework proposed in [23, 11] allows sensor-equipped mobile devices to report context information (e.g., traffic conditions, pollution reading) from their vicinity without risking their owners' location privacy. The system partitions the network domain into many tiles, each being a region that K users typically visit within a short time interval, and lets each node report its location at a granularity of tiles. It is unclear, though, how mobile nodes are updated with the latest tessellation information. Moreover, the proposed system assumes that each report is an independent event. In parallel to this work, a system [10] was proposed for privacy-preserving traffic monitoring based on the concept of *virtual trip lines* (VTLs). A VTL is a geographic marker that indicates where a vehicle needs to make a traffic report. For privacy protection, these markers are placed to avoid particularly sensitive areas. Their distances are also made large enough to prevent a user's consecutive location updates from being re-linked as a trajectory. This approach cannot be used for location privacy protection in LBSs because the placement of VTLs is predetermined.

7. CONCLUDING REMARKS

We have proposed a feeling-based model for location privacy protection in location-based services. The model allows a service user to express her privacy requirement by requesting that the location disclosed on her behalf must be at least as popular as some spatial region such as a shopping mall. Identifying such a region, called a *public region*, for privacy configuration is much more intuitive than specifying a number of K as in the traditional K-anonymity model. To measure the popularity of a spatial region, we borrow the concept of entropy from information theory to take into account not only the number of its visitors, but also the frequency of their visits. With this model in place, we investigate the problem of trajectory cloaking and propose a novel solution that is able to cloak a client's trajectory on the fly. The performance of the proposed technique is evaluated using both simulation and experiment. The prototype we implement can be used as a location privacy-aware gateway for users to entertain location-based services.

Our current techniques prevent an adversary from correlating anonymous location information with restricted spaces such as home and office to derive who was where at the time where the service was requested. In addition to such restricted space identification, other types of attack are likely in reality. One is observation implication attack. If an adversary has direct observation over the region where a user locates, the user does not have location privacy at that time point. However, the observed location may be linked to the user's future movement. Orthogonal to the observation implication is the exclusiveness attack. If the adversary knows that a user has never visited a certain region, then any trajectory which traverses through this region cannot belong to the user. These types of attacks may be prevented by applying the concept of l-diversity [24, 17] in trajectory cloaking, and we will investigate this in our future work.

8. REFERENCES

[1] xda-developers. http://wiki.xda-developers.com.

[2] TIGER/LINE CENSUS FILES. http://www.land.state.az.us/alris/doc/apendh.txt, 1990.

[3] A. R. Beresford and F. Stajano. Location Privacy in Pervasive Computing. In *IEEE Security and Privacy*, volume 2, pages 46–55, 2003.

[4] C. Bettini, X. S. Wang, and S. Jajodia. Protecting Privacy Against Location-Based Personal Indentification. In *Proceedings of the 2nd VLDB Workshop on Secure Data Management*, 2005.

[5] T. Brinkhoff. A Framework for Generating Network-Based Moving Objects. In *GeoInformatica*, volume 6(2), 2002.

[6] Y. Cai and T. Xu. Design, Analysis, and Implementation of a Large-scale Real-time Location-based Information Sharing System. In *ACM MobiSys'08*, pages 106–117, Breckenridge, Colorado, June 2008.

[7] R. Cheng, Y. Zhang, E. Bertino, and S. Prabhakar. Preserving User Location Privacy in Mobile Data Management Infrastructures. In *the 6th Workshop on Privacy Enhancing Technologies*, pages 393–412, 2006.

[8] C. Chow and M. F. Mokbel. Enabling Private Continuous Queries for Revealed User Locations. In *SSTD'07*, pages 258–275, 2007.

[9] C. Y. Chow, M. F. Mokbel, and X. Liu. A Peer-to-Peer Spatial Cloaking Algorithm for Anonymous Location-based Services. In *ACM GIS'06*, pages 171–178, November 2006.

[10] B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J. Herrera, A. Bayen, M. Annavaram, and Q. Jacobson. Virtual Trip Lines for Distributed Privacy-Preserving Traffic Monitoring. In *ACM Mobisys'08*, pages 15–28, June 2008.

[11] C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, and M. Shin. AnonySense: Privacy-Aware People-Centric Sensing. In *ACM Mobisys'08*, pages 211–224, June 2008.

[12] P. Samarati. Protecting Respondents' Identities in Microdata Release. In *IEEE TKDE*, volume 13(6), pages 1010–1027, 2001.

[13] B. Gedik and L. Liu. A Customizable k-Anonymity Model for Protecting Location Privacy. In *ICDCS'05*, pages 620–629, 2005.

[14] G. Ghinita, P. Kalnis, and S. Skiadopoulos. PRIVE: Anonymous Location-based Queries in Distributed Mobile Systems. In *Proc. of the 16th international conference on World Wide Web*, pages 371–380, Alberta, Canada, 2007.

[15] M. Gruteser and D. Grunwald. Anonymous Usage of Location-based Services through Spatial and Temporal Cloaking. In *ACM MobiSys'03*, pages 31–42, 2003.

[16] M. Gruteser and B. Hoh. On the Anonymity of Periodic Location Samples. In *Security in Pervasive Computing*, volume 3450/2005, pages 179–192, 2005.

[17] J. Han, T. Cen, and H. Yu. An improved v-mdav algorithm for l-diversity. *International Symposiums on Information Processing*, pages 733–739, 2008.

[18] Q. He, D. Wu, and P. Khosla. Personal Control over Mobile Location Privacy. In *IEEE Communications Magazine*, volume 42(5), 2004.

[19] B. Hoh and M. Gruteser. Protecting Location Privacy Through Path Confusion. In *IEEE/CreateNet Intl. Conference on Security and Privacy for Emerging Areas in Communication Networks*, pages 194–205, 2005.

[20] B. Hoh, M. Gruteser, H. Xiong, and A. Alrababy. Preserving Privacy in GPS Traces via Uncertainty-Aware Path Cloaking. In *ACM CCS'07*, pages 161–171, 2007.

[21] K. Ren, W. Lou, K. Kim, and R. Deng. A Novel Privacy Preserving Authentication and Access Control Scheme in Pervasive Computing Environments. In *IEEE Transactions on Vehicular Technology*, volume 55(4), 2006.

[22] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias. Preserving Anonymity in Location Based Services. In *Technical Report TRB6/06, Department of Computer Science, National University of Singapore*, 2006.

[23] A. Kapadia, N. Triandopoulos, C. Cornelius, D. Peebles, and D. Kotz. AnonySense: Opportunistic and Privacy-Preserving Context Collection. In *The Sixth International Conference on Pervasive Computing (PERVASIVE'08)*, pages 280–297, May 2008.

[24] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. L-diversity: Privacy Beyond K-Anonymity. *IEEE Transaction on Knowledge and Data Engineering*, 1(1), 2007.

[25] M. F. Mokbel, C.-Y. Chow, and W. G. Aref. The New Casper: Query Processing for Location Services without Compromising Privacy. In *Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB'06)*, pages 763–774, 2006.

[26] A. Inan and Y. Saygin. Location Anonymity in Horizontally Partitioned Spatial-Temporal Data. In *Master Thesis, Sabanci University, Turkey*, 2006.

[27] K. Sha, Y. Xi, W. Shi, L. Schwiebert, and T. Zhang. Adaptive Privacy-Preserving Authentication in Vehicular Networks (Invited Paper). In *Proceedings of IEEE International Workshop on Vehicle Communication and Applications*, 2006.

[28] C. Shannon. The Mathematical Theory of Communication. In *Bell System Technical Journal*, volume 30, pages 50–64, 1948.

[29] T. Xu and Y. Cai. Location Anonymity in Continuous Location-based Services. In *ACM GIS'07*, pages 300–307, November 2007.

[30] T. Xu and Y. Cai. Exploring Historical Location Data for Anonymity Preservation in Location-based Services. In *IEEE Infocom'08*, pages 547–555, Phoenix, AZ, 2008.

[31] 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 Service. In *ICDE'08*, pages 366–375, 2008.

[32] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.- L. Tan. Private Queries in Location-Based Services: Anonymizers are Not Necessary. In *ACM SIGMOD 2008*.

[33] A. Gkoulalas-Divanis, V. S. Verykios, and Bozanis P. A Network Aware Privacy Model for Online Requests in Trajectory Data. In *Data & Knowledge Engineering, DKE*, volume 68(4), page 431–452, April 2009.

[34] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private information retrieval. In *IEEE Symposium on Foundations of Computer Science*, pages 41–50, 1995.

[35] L. Sweeney. A Model for Protecting Privacy. In *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, volume 10(5), pages 557–570, 2002.

[36] G. Ghinita, P. Kalnis, and S. Skiadopoulos. MobiHide: A Mobilea Peer-to-Peer System for Anonymous Location-Based Queries. In *SSTD'07*, pages 221–238, 2007.