Private Queries and Trajectory Anonymization: a Dual Perspective on Location Privacy

Gabriel Ghinita

Dept. of Computer Science

Purdue University

West Lafayette, IN 47907, USA

E-mail: gghinita@cs.purdue.edu

Abstract. The emergence of mobile devices with Internet connectivity (e.g., Wi-Fi) and global positioning capabilities (e.g., GPS) have triggered the widespread development of location-based applications. For instance, users are able to ask queries about points of interest in their proximity. Furthermore, users can act as mobile sensors to monitor traffic flow, or levels of air pollution. However, such applications require users to disclose their locations, which raises serious privacy concerns. With knowledge of user locations, a malicious attacker can infer sensitive information, such as alternative lifestyles or political affiliations.

Preserving location privacy is an essential requirement towards the successful deployment of location-based services (LBS). Currently, two main LBS use scenarios exist: in the first one, users send location-based queries to an un-trusted server, and the privacy objective is to protect the location of the querying user. In the second setting, a trusted entity, such as a telephone company, gathers large amounts of location data (i.e., trajectory traces) and wishes to publish them for data mining (e.g., alleviating traffic congestion). In this case, it is crucial to prevent an adversary from associating trajectories to user identities. In this survey paper, we give an overview of the state-of-the-art in location privacy protection from the dual perspective of query privacy and trajectory anonymization. We review the most prominent design choices and technical solutions, and highlight their relative strengths and weaknesses.

1 Introduction

The increased popularity of mobile communication devices with embedded positioning capabilities (e.g., GPS) has generated unprecedented interest in the development of location-based applications. Consider the following scenario: Bob uses his GPS enabled mobile phone to ask the query "Find the nearest hospital to my present location". This query can be answered by a *Location-Based Service* (LBS) in a public server (e.g., Google Maps), which maintains a database with *points of interest (POI)*. However, the LBS is *not* trusted. To preserve his privacy, Bob does not contact the LBS directly. Instead he submits his query via an intermediate trusted server which hides his ID (services for anonymous web surfing are commonly available nowadays). However, the query still contains the exact coordinates of Bob. One may reveal sensitive user data, such as religious affiliations or alternative lifestyles, by combining the location with other publicly available information (e.g., a telephone directory).

Another interesting class of applications is the study of trajectory traces. Consider a company that offers integrated payment services: for instance, the *Octopus* [1] payment system deployed in Hong Kong enables users to pay for transportation and day-to-day purchases with a single proximity card. As a result, a large amount of transaction logs which contain movement data are gathered. A third party company or government organization may wish to access the data and derive trajectory patterns useful to optimize traffic flow. The data owner is bounded by contractual obligations to guarantee user privacy. On the other hand, releasing the data can provide significant revenues. The challenge is to publish trajectories in a privacy-preserving fashion that still allows the derivation of meaningful results (e.g., finding which road segments are most frequently subject to traffic jams).

The early work of Gruteser and Liu [12] identifies three aspects of location information disclosure: position awareness, sporadic queries and location tracking. Position awareness refers to the case where a device monitors an individual's location (e.g., an in-car GPS system), but no data is released to another party. The user's position is only used locally, to navigate a map for instance, hence no privacy threat occurs. The sporadic (or one-time) queries case refers to scenarios where a user reports his/her current location to a service provider, in order to find nearby points-of-interest (e.g., "find the closest restaurant"). Lastly, location tracking occurs in applications that require frequent updates of the user's position, e.g., traffic monitoring. Note that, these disclosure scenarios do not always occur separately. For instance, both sporadic and frequent location updates may arise in the case of private LBS queries. If a user issues a continuous query, e.g., "report the location of the closest restaurant while I move", multiple locations (or a tentative trajectory) must be sent to the service provider. On the other hand, the duration of location reporting may be much shorter than in the case when an automobile acts as a mobile sensor and reports its coordinates and velocity readings for the entire duration when the ignition is turned on.

Another important aspect in location disclosing is related to the attacker capabilities. In [12], the authors discuss the concepts of *weak* and *strong* privacy. Weak privacy requires that no sensitive data should be *directly* disclosed to a party that is not trusted. In other words, if the current location of the user does not reveal any sensitive information, it is safe to disclose. This requirement may be sufficient if an attacker only gains access to sporadic location updates. On the other hand, if the attacker has access to a history of locations, additional information can be inferred. For instance, if the trajectory of Bob includes along its way a hospital building, the attacker may associate him with a medical condition, even if Bob turns off his mobile device upon entering the hospital. In this case, *strong* privacy is required. Strong privacy disallows the publication of location snapshots which, although they do not represent a privacy violation by themselves, may be correlated to additional data to infer the presence of a user at a privacy-sensitive position. Anonymizing trajectory data is a representative example where strong privacy is necessary. Nevertheless, enforcing strong privacy must not have a significant negative impact on data accuracy, in the sense that the utility of the published data must be preserved.

In this paper, we provide a dual perspective on location privacy, by studying the two most prominent location-based application scenarios: private location queries and anonymous publication of trajectories. In Section 2, we outline the challenges that arise in protecting the privacy of users who issue LBS queries. We provide a taxonomy of solutions that achieve query privacy, and highlight their relative trade-offs with respect to privacy and performance. In Section 3, we survey methods that sanitize trajectory data before publication, in order to prevent associations between users and sensitive locations. We review techniques that publish independent location samples, as well as those that release individual trajectories. In Section 4, we conclude and identify several open issues that represent interesting directions for future research.

2 Private Location-Based Queries

To preserve privacy, the exact location of users that send queries to LBSs must not be disclosed. Instead, location data is first perturbed, or encrypted. For instance, some existing techniques generate a few random fake locations and send a number of redundant queries to the LBS [19, 28] to prevent user identification. Other methods employ the concept of k-anonymity [24, 26], a well-established concept in the publication of microdata (e.g., hospital records). In the LBS domain, *spatial k-anonymity (SKA)* is enforced by generating a *Cloaking Region (CR)* which includes the query source as well as $k-1$ other users [9, 11, 17, 22]. Finally, some techniques obscure the location data using spatial [18] or cryptographic [10] transformations.

Achieving privacy incurs an additional overhead in processing queries: for instance, a larger number of queries need to be processed in the case of techniques that generate redundant requests. For spatial k-anonymity techniques, query processing is performed with respect to the CR, which is considerably more expensive than processing point queries. Therefore, a trade-off emerges between privacy and performance.

We propose a taxonomy for LBS privacy techniques, consisting of three categories: *(1)* two-tier spatial transformations, *(2)* three-tier spatial transformations and *(3)* cryptographic transformations. Methods in Category 1 do not require any trusted third party, and the query anonymization is performed by the mobile user itself. Category 2 assumes the presence of a trusted third-party anonymizer server, and offers better protection against background knowledge attacks (e.g., an attacker may have additional information on user locations, from an external source). Category 3 offers the strongest privacy guarantees, and protects privacy even against powerful adversaries (e.g., attacker learns apriori all user locations).

2.1 Category 1: Two-Tier Spatial Transformations

Methods in this category involve only two parties at query time: the user and the LBS provider. Most of these methods[1] assume that no background knowledge is available to the attacker. A simple solution to query privacy is to generate a number of redundant queries for each real query. For instance, user u could generate r random "fake" locations, and send r redundant queries to the LBS, in addition to the actual query containing u's location. Such an approach is adopted in [19], where dummy locations are generated such that the resulting trajectories mimic realistic movement patterns. Dummy-generation algorithms can take into account movement parameters, such as velocity, and certain constraints, e.g., an underlying road network.

A more elaborate approach is *SpaceTwist* [28]: instead of generating a number of decoy locations beforehand, it performs a multiple-round, incremental nearest-neighbor query protocol, based on an *anchor* location. The anchor is initially set to a location randomly generated by the user. Throughout the query protocol, the user maintains two subsets of the dataspace: the *demand* and the *supply* spaces. The former consists of the space that needs

[1] A notable exception is the *PROBE* [7] system which assumes that the attacker knows all sensitive locations

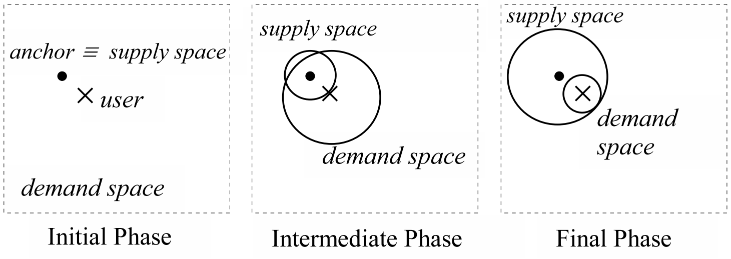


Figure 1: Incremental NN search process in SpaceTwist

to be covered by the issued queries, in order to ensure that the correct result is returned to the user, whereas the latter denotes the region of the space which is already covered. The client (the terms user and client are utilized interchangeably) knows both the demand space and the supply space, whereas the server knows only the supply space. Figure 1 gives an overview of query processing in SpaceTwist: initially, the demand space is set to the domain space, and the supply space contains only the anchor location. As points are retrieved from the server, the supply space expands. When a retrieved point is the closest point to the client seen so far, the results are updated, and the demand space shrinks. When the supply space eventually covers the demand space, it is termed final and the client is guaranteed to know its exact nearest-neighbor.

The more recent work in [18] uses the Hilbert curve mapping [4] to transform the dataspace of points of interest. In a pre-processing (off-line) stage, a trusted entity transforms each POI p_i into its Hilbert value $H(p_i)$, and uploads the values to the LBS. The parameters of the transformations (e.g., curve orientation, scale, etc), are kept secret from the LBS, and represent the encryption key. To allow encoding of queries and decoding of results, users possess tamper-resistant devices that store the encryption key. At query time, the user u computes its transformed location $H(u)$ and requests from the LBS the closest data value (in terms of 1D Hilbert values). Subsequently, the user decrypts the result by applying the inverse mapping H^{-1} to obtain the actual POI. The privacy of the solution relies on the large number of Hilbert curve parameter choices, and conjectures that it is computationally infeasible for the malicious attacker to decrypt Hilbert values to actual POI. Nevertheless, the above solution is approximate in nature and does not provide any guarantee on the result accuracy.

The *PROBE* [7] system introduces a novel approach to location privacy, by preventing the association between users and sensitive locations (similar to the $\ell - diversity$ [21] concept from microdata anonymization). In PROBE, it is assumed that the attacker has access to all sensitive locations from a particular data space (e.g., a city, a country, etc). Sensitive locations are represented by *features*, which are classified into *feature types* (e.g., hospitals, restaurants, etc). In an off-line phase, an *obfuscated map* is constructed, by partitioning the space into a set of disjoint regions such that the probability of associating each region to a certain feature type is bounded by a threshold. This process, called *obfuscation*, may require an additional trusted third party, but in the on-line phase (i.e., at query time) PROBE is a two-tier protocol. Figure 2 shows an obfuscated map with two obfuscated regions (OR_1 and OR_2): no region can be associated to the "hospital" feature type with probability higher than 44% (OC_1 contains 9 grid cells in total, 4 of which are sensitive, and $4/9 = 0.44$).

PROBE offers an amount of privacy which is superior to the other methods in this cat-

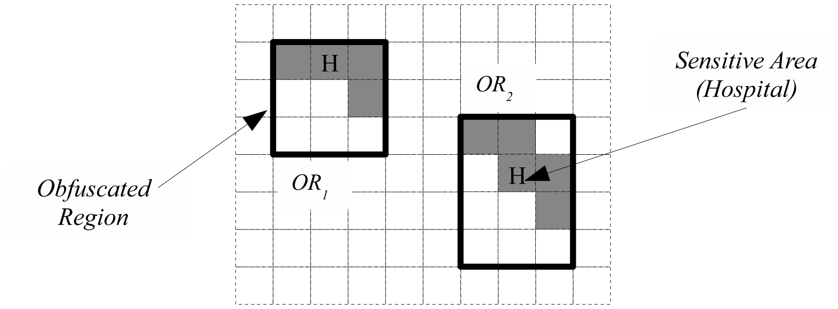


Figure 2: PROBE: Association Probability to "Hospital" Feature Type is Lower than 44%

egory. However, none of the two-tier spatial transformation solutions can prevent re-identification of the query source if an attacker has knowledge about specific user locations. For instance, if user u situated in a remote location issues a query (i.e., an outlier case), an attacker who knows that u is the only person residing in that area can associate u with the query, breaching user privacy. The next category of query anonymization methods deals with this issue.

2.2 Category 2: Three-Tier Spatial Transformations

Methods in this category implement the spatial k-anonymity paradigm: a cloaking region that contains $k-1$ users in addition to the query source (a k-CR) is generated, and the LBS processes the query with respect to the CR. Since all the k locations enclosed by the CR correspond to actual users (as opposed to "fake" locations in the previous category), the probability to identify the query source is at most $1/k$, even if the attacker has knowledge about exact user locations.

Most solutions in this category employ the three-tier architecture illustrated in Figure 3. A trusted centralized *anonymizer* acts as an intermediate tier between the users and the LBS. All users subscribe to the anonymizer and continuously report their location while they move. Each user sends his query to the anonymizer, which constructs the appropriate CR and contacts the LBS. The LBS computes the answer based on the CR, instead of the exact user location; thus, the response of the LBS is a superset of the answer. Finally, the anonymizer filters the result from the LBS and returns the exact answer to the user.

In *Casper* [22], the anonymizer indexes the locations of the clients using a pyramid data structure, similar to a Quad-tree. Assume u asks a query and let c be the lowest-level cell of the Quad-tree where u lies. If c contains enough users (i.e., $|c| \geq k$), c becomes the CR. Otherwise, the horizontal c_h and vertical c_v neighbors of c are retrieved. If $|c \cup c_h| \geq k$ or $|c \cup c_v| \geq k$, the corresponding union of cells becomes the CR; otherwise, the anonymizer retrieves the parent of c and repeats this process recursively. *Interval Cloak* [11] is similar to Casper in terms of both the data structure used by the anonymizer (a Quad-tree), and the cloaking algorithm. The main difference is that Interval Cloak does not consider neighboring cells at the same level when determining the CR, but ascends directly to the ancestor level. Casper and Interval Cloak guarantee privacy only for uniform distribution of user locations.

Hilbert Cloak [17] uses the Hilbert space filling curve to map the 2-D space into 1-D values. These values are then indexed by an annotated B$^+$-tree. The algorithm partitions the 1-D

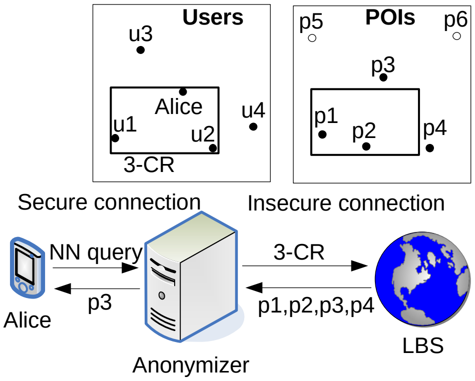


Figure 3: Spatial k-anonymity: Three-tier Architecture

sorted list into groups of k users (the last group may have up to $2k-1$ users). For querying user u the algorithm finds the group to which u belongs, and returns the minimum bounding rectangle of the group as the CR. The same CR is returned for any user in a given group. Hilbert Cloak guarantees privacy for any distribution of user locations, but only for one-time (i.e., single-snapshot) queries.

The previous approaches assume a static snapshot of user locations and do not consider correlation attacks (e.g., history of user movement). In [6], correlation attacks are handled as follows: At the initial timestamp t_0, cloaking region CR_0 is generated, which encloses a set AS of at least k users. At a subsequent timestamp t_i, the algorithm computes a new anonymizing region CR_i that encloses the same users in AS, but contains their locations at timestamp t_i. There are two drawbacks: *(i)* As users move, the resulting CR may grow very large, leading to prohibitive query cost. *(ii)* If a user in AS disconnects from the service, the query must be dropped. Furthermore, in [6] it is assumed that there are no malicious users.

Methods in Category 2 rely on the presence of other users to achieve spatial k-anonymity. These methods offer stronger privacy guarantees than Category 1 techniques, with the exception of PROBE. The privacy features of PROBE and spatial k-anonymity methods are not directly comparable: PROBE does not achieve k-anonymity, but it does provide spatial diversity. On the other hand, Category 2 techniques may not always prevent association of users to sensitive locations. For instance, it is possible for an entire CR to fall within a sensitive region (e.g., hospital). Therefore, the choice of paradigm (i.e., spatial anonymity vs spatial diversity) ultimately depends on the specific application requirements.

2.3 Category 3: Cryptographic Transformations

Recently, a novel LBS privacy approach based on *Private Information Retrieval (PIR)* was introduced in [10]. Two such methods are proposed, which support approximate and exact private nearest-neighbor search, respectively. PIR protocols [5, 20] allow a client to privately retrieve information from a database, without the database server learning what particular information the client has requested. Most techniques are expressed in a theoretical setting, where the database is an n-bit binary string X (see Figure 4). The client wants to find the value of the i^{th} bit of X (i.e., X_i). To preserve privacy, the client sends an encrypted request $q(i)$ to the server. The server responds with a value $r(X, q(i))$, which allows the client to compute X_i.

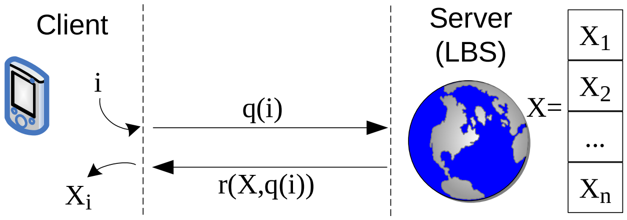
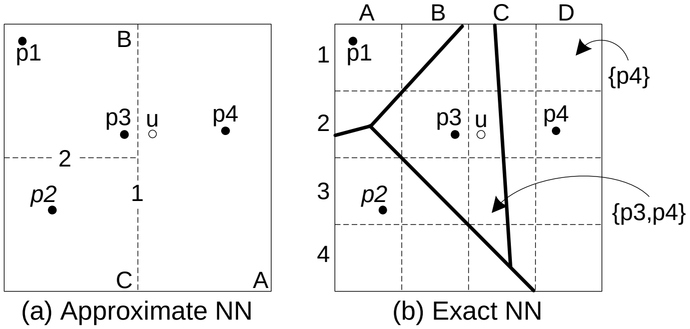


Figure 4: PIR Framework

The work in [10] builds upon the *computational* PIR ($cPIR$) protocol for binary data introduced in [20]. cPIR employs cryptographic techniques, and relies on the fact that it is computationally intractable for an attacker to find the value of i, given $q(i)$. Furthermore, the client can easily determine the value of X_i based on the server's response $r(X, q(i))$.

Note that, PIR protocols for binary data can support index-based queries, i.e. they retrieve the element with a given index i, whereas LBS queries are content-based, e.g. "find the closest POI to my location". The challenge in applying PIR to LBS privacy consists of finding effective methods to transform LBS queries into index-based queries. In [10] it is shown how to compute privately the nearest POI to a user location with acceptable cost, by retrieving a small fraction of the LBS' database. Figure 5.a outlines an approximative protocol, where u is the querying user and the LBS contains four points of interest p_1, p_2, p_3, p_4. In an off-line phase, the LBS generates a kd-tree index of the POIs and partitions the space into three regions A, B, C. To answer a query, the server first sends to u the regions A, B, C. The user finds the region (i.e., A) that contains him, and utilizes PIR to request all points within A; therefore, the server does not know which region was retrieved. The user receives the POIs in A in encrypted form and calculates p_4 as its NN. The method can be used with a variety of spatial indices. Note that, the result is approximate, since the true NN is p_3.

Figure 5.b outlines an exact NN protocol. In a pre-proces-sing phase, the server computes the Voronoi diagram [8] for the POI set. Each POI p_i is assigned to its Voronoi cell; by definition, p_i is the NN of any point within that cell. The server superimposes a regular grid of arbitrary granularity on top of the Voronoi diagram. Each grid cell stores information about the Voronoi cells intersecting it. For example $D1$ stores $\{p_4\}$, whereas $C3$ stores $\{p_3, p_4\}$. Upon asking a query, user u first retrieves the granularity of the grid, and calcu-



(a) Approximate NN

(b) Exact NN

Figure 5: Finding the Nearest Neighbor of u with PIR

lates the grid cell that contains u (i.e., $C2$). Then, u employs PIR to request the contents of $C2$. u receives $\{p_3, p_4\}$ (encrypted) and calculates p_3 as his exact NN.

2.4 The Privacy-Performance Trade-off

Supporting private LBS queries is achieved with an additional overhead in terms of computational and communication cost. The methods presented in this section provide various trade-offs between privacy and performance. Both aspects of this trade-off are discussed next.

2.4.1 The Privacy Aspect

Two-tier spatial transformations provide the least amount of privacy. For instance, dummy-generation offers no protection against attackers that possess background knowledge about user locations. Furthermore, exact locations are disclosed to the LBS, which is undesirable, since an attacker can learn that one of these locations corresponds to the actual user. SpaceTwist achieves slightly better privacy protection, because it does not disclose exact user locations. Still, an attacker that has knowledge on the user distribution within the supply space (see Section 2.1) could infer the identity of the query source.

PROBE offers the strongest privacy features among the methods in Category 1. It assumes that the attacker has knowledge on all sensitive feature locations, and prevents the user-sensitive location association. In the worst case, however, an attacker may be able to associate a user with a sensitive query, since an obfuscated region may contain no other users in addition to the query source.

Methods in Category 2 (three-tier spatial transformations) do prevent user re-identification, because cloaked regions contain at least k actual users. Still, a cloaking region can have reduced extent (in the worst case, it can degenerate to a point). Therefore, an attacker may learn the whereabouts of the query source (and a number of $k-1$ other users), and associate the locations of all users in the CR with a sensitive feature, compromising their privacy. No direct comparison can be made between PROBE, which offers spatial diversity, and Category 2 methods that provide spatial anonymity. The choice of one method over another depends on the privacy requirements of each specific scenario. Furthermore, none of the Category 1 or 2 methods is able to protect user privacy for continuous queries (i.e., over multiple snapshots of locations).

The PIR-based method in [10] provides full-featured privacy, under all attack models, since no information about user location is disclosed. Therefore, no association between users and sensitive locations can be performed. Even in the case when the attacker knows the exact locations of all users (from an external source), the association between users and queries is still prevented. The privacy guarantees hold for continuous queries (i.e., moving users) as well.

2.4.2 The Performance Aspect

The main overhead incurred by the dummy-generation scheme consists of processing redundant queries. However, there is only one round of processing. SpaceTwist also requires the processing of several point queries, but the process consists of multiple user-server interaction rounds, which can lead to increased response time.

PROBE and spatial anonymity methods incur a similar overhead in terms of query processing: a region query is processed in each case (corresponding to either a cloaked, or

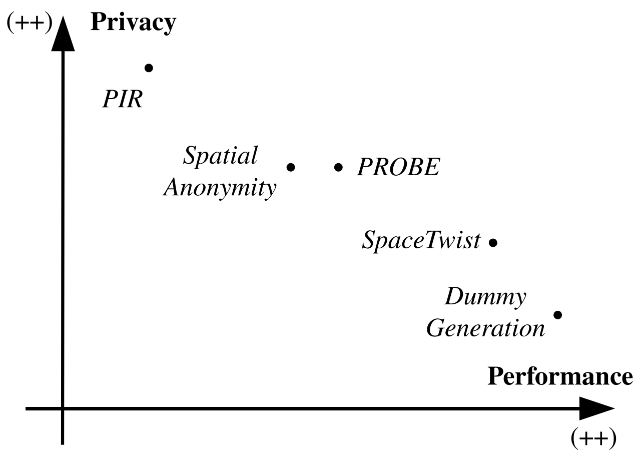


Figure 6: Privacy-Performance Tradeoff in LBS Privacy

an obfuscated region). Depending on the particular application scenario (e.g., density of sensitive locations and density of users), the relative performance of the two types of methods may vary. However, PROBE has the additional advantage that at query time, there is no overhead associated to obfuscated region generation, which is done off-line. In contrast, spatial anonymity methods require on-line cloaked region generation and frequent updates of user locations.

Finally, the PIR cryptographic-based approach in Category 3 may incur significant processing overhead, linear to the number of POI. As shown in [10], performance can be improved through a number of optimizations, such as re-using partial computation results, and parallelization. Still, the overhead is likely to exceed that of spatial transformation methods. Figure 6 shows a graphical representation of the discussed methods with respect to the privacy-performance trade-off achieved.

3 Trajectory Anonymization

Figure 7 shows the typical scenario of trajectory anonymization: a central trusted component (e.g., telecom company) gathers location samples from a large number of subscribers. The collected data is then shared with other un-trusted entities for various purposes, such as traffic optimization research. The trusted entity is the data *publisher*, and must ensure that releasing the data does not compromise user privacy. The publisher is assumed to be bound by contractual obligations to protect the users' interests, therefore it is trusted not to allow privacy breaches to occur. Location collection is performed in an on-line fashion. Anonymization, on the other hand, is likely to be a more costly operation, and can be performed off-line (although on-line anonymization is also possible).

When trajectory[2] data is published, a malicious attacker may associate user identities to sensitive locations, thus compromising users' privacy. One straightforward solution to protect privacy is to remove all user identifiers from the data, or replace them with pseudo-identifiers. The work in [3] proposes the use of *mix zones*, i.e., areas with high user density where many paths intersect. The idea is to confuse adversaries by inter-changing pseudo-identifiers among users within the mix zones. However, such an approach is not suffi-

[2] We use the terms *trajectory* and *track* interchangeably.

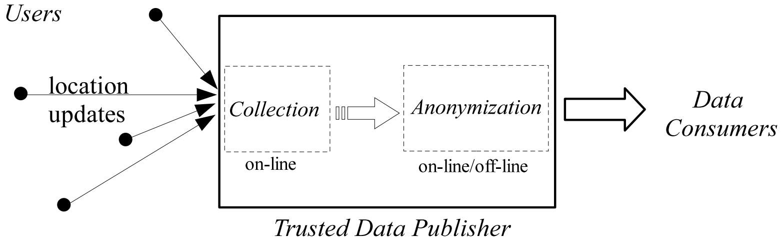


Figure 7: Trajectory Publication: System Architecture

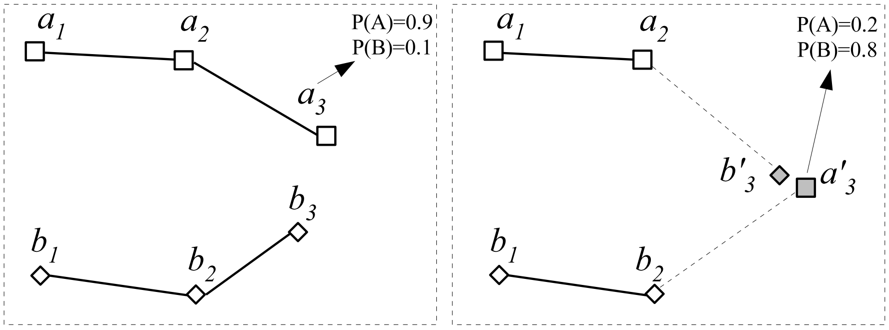
cient against sophisticated adversaries who may possess background knowledge (e.g., address of a user's residence) and may employ advanced target-tracking algorithms [23] to re-construct trajectories from successive location samples.

Privacy-preserving trajectory publication techniques can be broadly classified into two categories: methods that publish independent location samples, and methods that publish individual trajectories. Techniques in the former class are suitable for applications where only aggregate location data is required, e.g., traffic monitoring. Such methods are reviewed in Section 3.1. Solutions in the latter category publish individual trajectories, but distort location samples at each timestamp. These methods are reviewed in Section 3.2, and are suitable for applications where the causality relationship between source and destination locations is important.

3.1 Publishing Independent Location Samples

The early work of [12] proposes three algorithms to prevent disclosure of sensitive location data. All three assume the existence of a map with the sensitive locations in the enclosing geographical region. The first solution, called *base*, simply suppresses from publication those updates corresponding to sensitive locations. This method achieves only weak privacy. The second solution, called *bounded-rate*, aims to achieve strong privacy by suppressing with fixed frequency updates in non-sensitive areas (in addition to withholding sensitive locations). The idea is to filter out some of the non-sensitive locations that may represent entry points to sensitive areas. Finally, the third method, called *k-area*, splits the map into zones, or cells, that have both sensitive and non-sensitive locations. All updates in a given cell are delayed until the user exits that particular cell. Subsequently, the previous cell locations are disclosed only if the user had not visited any sensitive location in that cell. None of these three solutions protects against an attacker with background knowledge.

Later in [14], the same authors show that multiple-hypotheses tracking algorithms (MTT) [23] based on Kalman filters are very successful in re-constructing trajectories (in some cases, by matching close to 100% of location samples to the correct path). Once the trajectories are re-constructed, they can be mapped to user identities, based on a small number of known samples (e.g., a user's home or office). In addition, the success probability of identifying tracks and matching users to tracks increases with the duration of track acquisition. Figure 8(a) shows an example with two user trajectories $A : a_1 - a_2 - a_3$ and $B : b_1 - b_2 - b_3$. Even if user identifiers are suppressed, and location samples are independently published for each timestamp (i.e., as pairs $(a_1, b_1) \cdots (a_3, b_3)$), filtering allows an attacker to assign sample a_3 to track A with 90% probability. To confuse the attacker, individual location samples are distorted, in order to minimize the probability of successfully matching loca-



(a) Original Location Data (b) Perturbed Location Data

Figure 8: Path Perturbation Prevents Trajectory Re-Construction

tions to trajectories. For instance, in Figure 8(b), the locations of samples a_3 and b_3 are changed to a'_3 and b'_3, respectively. The attacker assigns location a'_3 to track B with high probability (0.8). Therefore, track matching for the third timestamp (and likely subsequent timestamps as well) is prevented.

Distorting location samples inherently introduces data inaccuracy, and may impact correct query processing on top of the data. A trade-off among privacy and accuracy emerges. The authors of [14] propose metrics to quantify both privacy and inaccuracy. Specifically, privacy is measured by the *expectation of distance error*, which captures how accurate an adversary can match locations to tracks. Given N users (hence N location samples at each timestamp) and an observation time of M timestamps, expectation of distance error for trajectory of user u is measured as

$$E[u] = \frac{1}{NM}\sum_{i=1}^{M}\sum_{j=1}^{I_i} p_j(i)d_j(i)$$

where I_i is the total number of assignment hypotheses for user u at timestamp i, $p_j(i)$ is the probability associated with hypothesis j at timestamp i, and $d_j(i)$ is the distance between the actual and estimated position of u at timestamp i. The data accuracy is measured according to the quality of service (QoS) metric

$$QoS = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M}\sqrt{(x_{u_i}(j) - x'_{u_i}(j))^2 + (y_{u_i}(j) - y'_{u_i}(j))^2}$$

where (x_{u_i}, y_{u_i}) and (x'_{u_i}, y'_{u_i}) are the actual and perturbed coordinates of u_i, respectively.

The data undergo a *Path Perturbation* phase, which formulates the problem of confusing the attacker as a constrained non-linear optimization problem. The objective is to maximize the privacy function E under the constraint that the maximum distortion for each individual published location does not exceed a threshold R, which is application-dependent. The perturbation phase needs to consider all permutations of assigning location samples to tracks, hence the computational cost is very high. For instance to perform perturbation for N user trajectories of M samples each, the complexity is $O(N!)^M$, which is not feasible in practice. To decrease the overhead, a *Path Segmentation* phase is performed prior to perturbation. The idea is to reduce the search space for the constraint optimization problem by pruning some of the most unlikely hypotheses of assigning samples to tracks.

The Path Perturbation algorithm maximizes the privacy metric under the given accuracy constraint R. However, this is not sufficient to protect the privacy of users in sparse areas. If user trajectories are situated far apart from each other, even the best achievable expectation of distance error may not be enough to prevent re-identification. The work in [16] acknowledges this limitation and proposes a solution that relies on the k-anonymity concept. However, the authors observe that simple location cloaking, as used for private queries, is not suitable for publishing trajectory data, because it severely distorts data. In traffic monitoring applications, for instance, in order to match a user location to a road segment with high probability, the spatial accuracy must be within $100m$. It is shown experimentally that k-anonymization based spatial cloaking fails to achieve this threshold: for instance, with a real trajectory trace and a relatively low anonymity degree ($k = 3$), the obtained accuracy is $500m$.

In practice, the privacy threat occurs when individual trajectories can be associated with particular users. Furthermore, such association can not be performed in very dense areas, but only in sparse areas, and the attacker's success probability increases with the length of the disclosed trajectory. Based on these observations, the work in [16] introduces two new privacy metrics. The first one is called *Time-to-Confusion (TTC)*, and measures the maximum number of consecutive timestamps for which locations along the same trajectory are safe to disclose. If TTC is exceeded for a certain track, its subsequent location samples are suppressed, to prevent trajectory re-construction. The second metric used is *tracking uncertainty*, which measures for each user u the entropy

$$H(u) = -\sum_i p_i \log p_i$$

where p_i is the probability of associating u to the location sample i in a particular snapshot. The attacker's *confidence* is $1 - H$. An algorithm is proposed which verifies at each timestamp if the TTC and the attacker's confidence are below certain thresholds. If thresholds are exceeded, the corresponding location samples are suppressed. The work in [15] follows a similar approach, and extends the solution in [16] by allowing location updates only at well-specified points along a *Virtual Trip Line (VTL)*. The points along each VTL are chosen in such a manner that they correspond to privacy-insensitive locations only.

3.2 Publishing Individual Trajectories

The objective of the techniques presented in the previous section is to prevent an attacker from re-constructing trajectories based on independent locations. Publishing independent location samples is useful for applications that require only aggregate information, such as traffic monitoring. However, in other classes of applications, the movement patterns and the causality relationship between certain source and destination locations may be of interest. In such cases, it is necessary to publish in a privacy-preserving manner entire trajectories, rather than independent location samples. Preventing a malicious adversary from correlating location samples is no longer a challenge. Instead, the focus is on perturbing trajectory data to prevent the association of individuals with trajectories.

The work in [27] considers a scenario where location samples are drawn from a discrete set (e.g., retail points, tourist attractions, etc.), and assumes an attack model with clearly-defined background knowledge. Specifically, the attacker already knows some trajectory fragments, and the identity of users corresponding to those fragments. Consider the example of a company P (publisher) that commercializes cards as a convenient form of payment.

Original Data

t_1	a_1-a_2-a_3-b_1
t_2	a_1-a_2-b_2
t_3	a_1-a_3-b_1
t_4	a_1-a_3-b_2

A's View

u_1	a_1-a_2-a_3
u_2	a_1-a_2
u_3	a_1-a_3
u_4	a_1-a_3

Transformed Data

t'_1	a_1-a_2-b_1
t_2	a_1-a_2-b_2
t_3	a_1-a_3-b_1
t_4	a_1-a_3-b_2

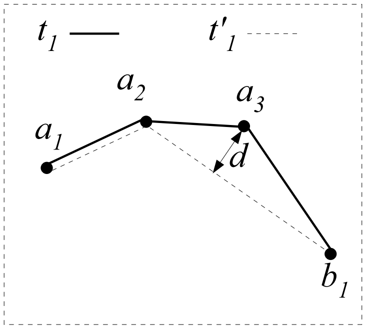
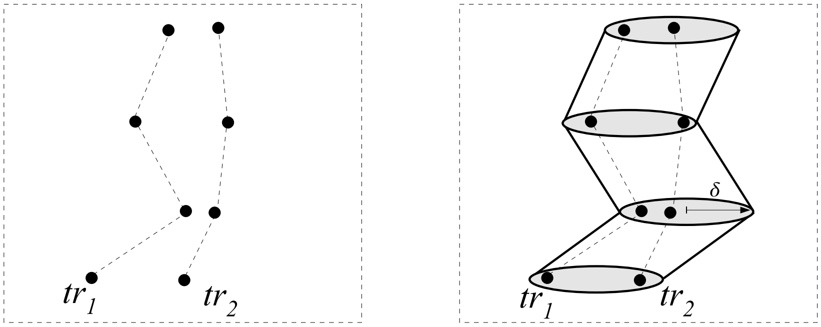


Figure 9: Protecting against Attackers that Know Trajectory Sub-Sequences

Such cards can be used to pay for transportation, as well as for day-to-day purchases. In time, P (which is trusted by all card users) will gather large amounts of trajectory data, which can be useful for a variety of purposes (e.g., inferring consumer travel and spending patterns). However, P is required by law not to compromise the privacy of its customers. Furthermore, the partner companies of P are not trusted. For example, a retail chain company A has access to all purchases by user u, and also learns the identity of u through a customer-fidelity program. Thus, A has access to a sub-set of the trajectories followed by u, and may wish to find out what other locations u has visited. P must prevent against this sort of privacy threat.

Consider the example of Figure 9, where P publishes the original location data. Partner companies A and B have knowledge about subsets of trajectories corresponding to their points of operation. Such known locations are denoted by a_i and b_j, respectively. A may try to infer the other locations that its customers have visited, by inspecting the original data. For instance, A can identify that u_1 corresponds to trajectory t_1, since only t_1 matches the $a_1 - a_2 - a_3$ movement pattern known by A. Therefore, A can infer with certainty that u has visited location b_1, which may correspond to a nightclub. Such a sensitive association is clearly a privacy breach. The fact that allowed the staging of a successful attack was A's ability to identify the trajectory of u_1. This situation is similar to that of publishing microdata, such as hospital records [24, 26]. In trajectory publication, the location data known to the attacker (A's view) is used as a quasi-identifier, whereas the location samples corresponding to other companies (e.g., B) are sensitive attributes. The authors of [27] approach the trajectory anonymization problem from a data mining [13] perspective. Specifically, each trajectory is regarded as a *transaction*, and each location sample represents a transaction *item*, according to data mining terminology. A particular *itemset* that is present in the attacker's database can be used as a quasi-identifier. To prevent re-identification of trajectories, the published data must consist of transactions in which every itemset occurs a sufficient number of times. In other words, the *support* [13] of each itemset must be large enough to ensure that the privacy breach probability P_{br} is below a threshold. Privacy breach is quantified as the probability of associating an individual trajectory with a particular sensitive location/item, e.g., b_1 in the earlier example. Before publication, the data is sanitized through a greedy heuristic that suppresses individual location samples. In Figure 9, the value a_3 is removed from the original data. As a result, A can only infer that the trajectory of u_1 is other t'_1 or t_2. Therefore, u_1 may have visited either b_1 or b_2 with equal



Original Trajectories

2-Anonymous Trajectory "Tube"

Figure 10: Anonymizing Trajectories through Space Translation

likelihood, hence the breach probability is reduced from 100% to 50%. The inherent inaccuracy in removing samples is measured by the path deviation introduced. In our example, by removing a_3, the obtained deviation is $dist(t_1, t_1') = d$.

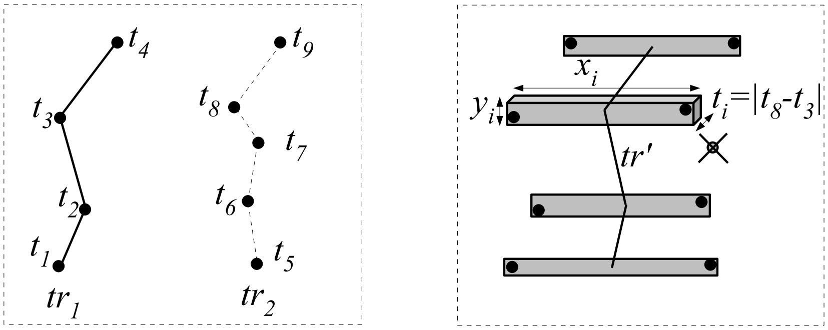
The work in [2] also employs the concept of k-anonymity for trajectories. However, as opposed to [27], it considers continuous-space location samples. Privacy is mainly enforced through location *generalization*, rather than suppression as in the case of [27]. The privacy paradigm proposed by [2] is (k, δ)-anonymity, which is achieved using the concept of *anonymizing tubes*, i.e., a sequence of cylindrical volumes that spatially enclose user trajectories. Figure 10 shows an example of $k = 2$ *co-located* trajectories, i.e., trajectories that can be enclosed by a tube with radius δ. It is considered that trajectories are defined over the same discrete time domain $[t_1 \cdots t_n]$, and $\forall t \in [t_1 \cdots t_n]$ and any two trajectories τ_1, τ_2 inside an anonymized tube, it holds that

$$dist((\tau_1[t].x, \tau_1[t].y)(\tau_2[t].x, \tau_2[t].y)) \leq \delta$$

where $\tau[t].x$ and $\tau[t].y$ represent the coordinates of the location sample along trajectory τ at time t.

For each group of trajectories, the polygonal line that represents the center of the tube is published. The data distortion is measured by summing over all timestamps the distance between the location sample of each trajectory and the cylinder center at that particular timestamp. Anonymization is performed through a two-step algorithm. First, a pre-processing phase is employed, to horizontally partition the set of trajectories according to their timeframes (i.e., obtain disjoint sets of trajectories that are concomitant). Next, a greedy clustering heuristic is performed within each partition, to obtain groups of at least k trajectories each. The tube that encloses each group is published.

In [25], trajectory privacy is also achieved by employing k-anonymity. However, as opposed to [2], trajectories from distinct timeframes can be anonymized together, and their time difference is factored in the data inaccuracy metric used. An algorithm for trajectory clustering in the three-dimensional space-time domain is proposed, which creates groups of at least k transactions each. Trajectories belonging to the same group are generalized, such that they are indistinguishable from each other. Grouping involves generalization of both spatial and temporal coordinates, i.e., all trajectories in the same group are replaced with their spatio-temporal bounding box. An example of anonymizing two trajectories, tr_1 and tr_2, is shown in Figure 11. Each location sample is tagged with the timestamp at which it was collected. The resulting bounding boxes together with their time differences (i.e.,



Original Trajectories

2-Anonymous Group

Figure 11: Anonymizing Trajectories through Spatio-Temporal Generalization

the time period covered by each bounding box) incur information loss, which is measured according to a *log-cost metric (LCM)*. LCM quantifies the trajectory inaccuracy in both space and time, and is calculated by summing the enlargement required over each of the M published snapshots of locations. A weighting factor for space (w_s), as well as time (w_t) can be specified, depending on the application that uses the data. Formally,

$$LCM = \sum_{i=1}^{M} [w_s(\log |x_i| + \log |y_i|) + w_t \log |t_i|]$$

Anonymization of trajectories is performed in two stages. First, the algorithm chooses the trajectories that will belong to each group. This phase is performed through an heuristic that is similar to string matching. Next, an anonymization phase is performed, where it is decided which samples from each trajectory will be anonymized with samples from other trajectories. Note that, not all trajectories have the same number of samples. Furthermore, not all location samples must be retained: for instance, in Figure 11, the sample with timestamp t_7 from tr_2 is suppressed.

4 Conclusions

Location privacy has already been acknowledged as an important problem, and effective privacy-preserving solutions will be necessary to support the widespread development and adoption of location-based applications. The current survey paper identified two main facets of location privacy, and overviewed the state-of-the-art in private location queries and anonymous trajectory publication.

For the private queries domain, we have proposed a taxonomy of anonymization techniques and highlighted their relative privacy-performance trade-offs. At one end of the spectrum, methods such as SpaceTwist and dummy-generation schemes incur low overhead, but they only provide privacy under a limited set of assumptions (i.e., attacker has no background knowledge). At the opposite end, the PIR approach from [10] offers strong privacy even in highly-adversarial scenarios. Finally, the spatial anonymity (Category 2) and spatial diversity (PROBE) techniques manage to provide a good amount of privacy under reasonable attack assumptions, and incur moderate overhead.

The research area of privacy-preserving trajectory publication is currently fractured between two different paradigms: releasing independent location samples versus publishing entire trajectories. The advocates of the former approach claim that generalization and suppression of data, which is inherent if individual trajectories are published, cannot achieve the level of accuracy dictated by practical applications. On the other hand, publication of independent samples may not be sufficient if complex tasks (e.g., deriving travel patterns) are performed on top of the data.

There are several interesting open directions for location privacy research. For the private queries problem, it is interesting to study combined methods that achieve both anonymity and diversity (i.e., they prevent both user-sensitive location and user-sensitive query associations). For instance, the obfuscation map construction of PROBE could be augmented with an on-line component that verifies that sufficient users are included in an obfuscated region. Another promising direction is to develop PIR-based techniques that support more complex types of queries. It is also worth investigating methods that can further reduce the overhead of PIR. One possible approach would be to design a hybrid method that combines spatial transformations (at a coarser level of data space granularity), followed by a PIR phase on a restricted portion of the data space.

For trajectory publication, it would be interesting to study if location privacy can be achieved independently of a centralized trusted entity. Storing all location data in one place may represent a privacy threat in itself, if the centralized publisher is compromised. A more suitable approach may have users anonymize their tracks in a collaborative manner. Finally, it may be interesting to find whether cryptographic techniques could be applied to trajectory anonymization as well. In this setting, query processing could be performed directly on the encrypted data, and the assumption of an intermediate trusted entity can be relinquished.

References

[1] *The Octopus Payment System: http://www.octopuscards.com*.

[2] O. Abul, F. Bonchi, and M. Nanni. Never walk alone: Uncertainty for anonymity in moving objects databases. In *ICDE*, pages 376–385, 2008.

[3] A. R. Beresford and F. Stajano. Mix zones: User privacy in location-aware services. In *IEEE Annual Conference on Pervasive Computing and Communications Workshops*, 2004.

[4] A. R. Butz. Alternative Algorithm for Hilbert's Space-Filling Curve. *IEEE Trans. Comput.*, C-20(4):424–426, 1971.

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

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

[7] M. Damiani, E. Bertino, and C. Silvestri. PROBE: an Obfuscation System for the Protection of Sensitive Location Information in LBS. Technical Report 2001-145, CERIAS, 2008.

[8] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 2nd edition, 2000.

[9] B. Gedik and L. Liu. Location Privacy in Mobile Systems: A Personalized Anonymization Model. In *Proc. of ICDCS*, pages 620–629, 2005.

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

[11] M. Gruteser and D. Grunwald. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In *Proc. of USENIX MobiSys*, 2003.

[12] M. Gruteser and X. Liu. Protecting Privacy in Continuous Location-Tracking Applications. *IEEE Security and Privacy*, 2:28–34, 2004.

[13] J. Han. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., 2005.

[14] B. Hoh and M. Gruteser. Protecting location privacy through path confusion. In *Proc. of International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM)*, pages 194–205, 2005.

[15] B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J.-C. Herrera, A. M. Bayen, M. Annavaram, and Q. Jacobson. Virtual trip lines for distributed privacy-preserving traffic monitoring. In *Proc. of the 6th international conference on Mobile systems, applications, and services (MobiSys)*, pages 15–28, 2008.

[16] B. Hoh, M. Gruteser, H. Xiong, and A. Alrabady. Preserving privacy in GPS traces via uncertainty-aware path cloaking. In *Proc. of the 14th ACM conference on Computer and Communications Security (CCS)*, pages 161–171, 2007.

[17] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias. Preserving Location-based Identity Inference in Anonymous Spatial Queries. *IEEE TKDE*, 19(12), 2007.

[18] A. Khoshgozaran and C. Shahabi. Blind Evaluation of Nearest Neighbor Queries Using Space Transformation to Preserve Location Privacy. In *SSTD*, 2007.

[19] H. Kido, Y. Yanagisawa, and T. Satoh. An anonymous communication technique using dummies for location-based services. In *International Conference on Pervasive Services (ICPS)*, pages 88–97, 2005.

[20] E. Kushilevitz and R. Ostrovsky. Replication is NOT needed: Single database, computationally-private information retrieval. In *IEEE Symposium on Foundations of Computer Science*, pages 364–373, 1997.

[21] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. l-Diversity: Privacy Beyond k-Anonymity. In *ICDE*, 2006.

[22] M. F. Mokbel, C. Y. Chow, and W. G. Aref. The New Casper: Query Processing for Location Services without Compromising Privacy. In *Proc. of VLDB*, 2006.

[23] D. Reid. An Algorithm for Tracking Multiple Targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, 1979.

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

[25] Y. Saygin, E. Nergiz, and M. Atzori. Towards trajectory anonymization: a generalization-based approach. In *International Workshop on Security and Privacy in GIS and LBS (SPRINGL)*, pages 52–61, 2008.

[26] L. Sweeney. k-Anonymity: A Model for Protecting Privacy. *Int. J. of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.

[27] M. Terrovitis and N. Mamoulis. Privacy preservation in the publication of trajectories. In *Proc. of International Conference on Mobile Data Management(MDM)*, pages 65–72, 2008.

[28] M. L. Yiu, C. Jensen, X. Huang, and H. Lu. SpaceTwist: Managing the Trade-Offs Among Location Privacy, Query Performance, and Query Accuracy in Mobile Services. In *International Conference on Data Engineering (ICDE)*, pages 366–375, 2008.