Trajectory Anonymity in Publishing Personal Mobility Data

Francesco Bonchi
Yahoo! Research
Barcelona, Spain
bonchi@yahoo-inc.com

Laks V.S. Lakshmanan
University of British Columbia
Vancouver, BC, Canada
laks@cs.ubc.ca

Hui (Wendy) Wang
Stevens Institute of Technology
Hoboken, NJ, USA
Hui.Wang@stevens.edu

ABSTRACT

Recent years have witnessed pervasive use of location-aware devices such as GSM mobile phones, GPS-enabled PDAs, location sensors, and active RFID tags. The use of these devices generates a huge collection of spatio-temporal data, variously called moving object data, trajectory data, or moblity data. These data can be used for various data analysis purposes such as city traffic control, mobility management, urban planning, and location-based service advertisements. Clearly, the spatio-temporal data so collected may help an attacker to discover personal and sensitive information like user habits, social customs, religious and sexual preferences of individuals. Consequently, it raises serious concerns about privacy. Simply replacing users' real identifiers (name, SSN, etc.) with pseudonyms is insufficient to guarantee anonymity. The problem is that due to the existence of quasi-identifiers, i.e., spatio-temporal data points that can be linked to external information to re-identify individuals, the attacker may be able to trace the anonymous spatio-temporal data back to individuals.

In this survey, we discuss recent advancement on anonymity preserving data publishing of moving object databases in an off-line fashion. We first introduce several anonymity models, then we describe in detail some of the proposed techniques to enforce trajectory anonymity, discussing their merits and limitations. We conclude by identifying challenging open problems that need attention.

1. INTRODUCTION

The discussion on mobility data collection and the associated privacy issue, has revamped in the last few months thanks to various initiatives, and several articles, e.g., on The New York Times[1] or Wired[2], discussing the fact that indeed, mobile phone producers and service providers have been collecting users' locations continuously, no matter whether the users volunteer for this collection or not.

The debate has reached the political spheres in many countries. A German Green party politician, Malte Spitz, went to court to find out exactly what his cellphone company, Deutsche Telekom, knew about his whereabouts. The results were astounding. In a six-month period – from September, 2009, to the end of February 2010 – Deutsche Telekom had recorded and saved his longitude and latitude coordinates more than 35,000 times, tracing him exactly during his trips and at home.[3]

Discussing to which extent the whereabouts of a person represent a private and sensitive information, is a philosophical, social and legal issue which is beyond the scope of this paper. It is interesting to note that for many individuals nowadays this is not an issue at all, on the contrary, updating continuously their contacts in Facebook or Twitter about their whereabout is a common practice. Just think about Foursquare[4], a platform which has turned reporting location updates into a social game.

In this paper we give for granted that the collected location data may help a malicious attacker to discover personal and sensitive information like user habits, social customs, religious and sexual preferences of individuals. We do not even consider the privacy issues related to the data collection itself, instead we focus on how to anonymize mobility data in the case that, for analysis purposes, they must be published, i.e., shared with some 3rd party.

In fact, mobility data can be used for various data analysis based applications such as city traffic control, mobility management, urban planning, and location-based service advertisements, just to mention a few.

The extent of research effort on these data is evidenced by the number of spatio-temporal data mining techniques that have been developed in the last several years [28; 29; 27; 42; 43; 49; 36; 58; 44; 9].

Therefore it is important to develop techniques to transform a database of trajectories of moving objects, in such a way that it satisfies some concept of anonymity, while most of its original utility is maintained in the transformed database.

Simply replacing users' real identifiers (e.g., name, SSN, etc.) with pseudonyms is insufficient to guarantee anonymity. The problem is that due to the existence of the *quasi-identifier* locations, i.e., a set of locations that can be linked to external information to re-identify individuals, the anonymous location data may be traced back to personally identifying information with the help of additional data sources.

As an example, [41] shows that based on two-week GPS

[1] http://www.nytimes.com/2011/03/26/business/media/26privacy.html?_r=1&_r=2\&smid=tw-nytimes\&seid=auto

[2] http://www.wired.com/gadgetlab/2011/04/apple-iphone-tracking/

[3] http://www.zeit.de/digital/datenschutz/2011-03/data-protection-malte-spitz

[4] https://foursquare.com/

tracks from 172 known individuals, the home address (with median error below 60 meters) and identity (with success above 5%) of these individuals were successfully identified by joining GPS traces with a reverse geocoder and a Web-based whitepage directory.

The problem of location privacy has been well studied in the context of location-based services [39; 46; 31; 22; 47], but with a focus on *on-line*, *service-centric* anonymity. In this paper, we consider *off-line* and *data-centric* anonymity, as in the context of data publishing. In particular, we consider the following publishing framework. A central trusted component (e.g., telecom company) gathers trajectories from a large number of users. The trusted entity publishes the collected trajectories to share with other entities, which may be un-trusted, for various purposes, such as traffic optimization research. The publisher is *assumed* to apply privacy-preserving techniques on the mobility data so that no privacy breach will occur w.r.t. assumed attack models.

In this survey, we discuss the recent advancement on anonymity preserving data publishing of personal mobility databases in an off-line fashion, that is, when we want to publish a stable database of moving objects.

We first categorize the adversary knowledge in the current work into two types: (1) adversary knowledge based on location only, and (2) adversary knowledge based on mobility patterns. The location-based adversary knowledge is applied on the *original* trajectory database; the adversary will identify individuals by matching the known locations to the published trajectories. The mobility pattern based adversary knowledge is mainly applied on the *anonymized* trajectories; given the cloaked trajectories which may be robust against location-based attacks, the attacker will use mobility patterns to predict the real locations and de-generalize the anonymized trajectories, and thus bring privacy vulnerabilities.

For each category, we present in detail some of the most prevalent methodologies that have been proposed recently, including the definition of adversary knowledge, the privacy model, and anonymization techniques. Table 1 reports the classification of papers adopted in this survey.

	Adversary Knowledge	
	Location Based	Motion Pattern Based
QID-aware	[56], [57]	[24], [37]
QID-blind	[2; 3], [45], [50]	

Table 1: A classification of the main papers reviewed in this survey.

Paper content and organization. In Section 2, we review the most common application scenarios of publishing trajectory databases and the privacy issues in these scenarios. Then we focus on a few papers (all very recent) that, to the best of our knowledge, are the unique that have attacked the problem of trajectory anonymization so far. We categorize these papers into two types, viz., methods based on location-based adversary knowledge (Section 3) and those based on mobility pattern based adversary knowledge (Section 4). Finally, in Section 5, we draw some conclusions and discuss important open research problems.

2. APPLICATION SCENARIOS AND PRIVACY ISSUES

GPS Traces. The personal mobility databases that have caught most attention are those that are collected from GPS-based devices [2; 35]. In these datasets, first, the personal trajectories are normally continuous, since the moving objects can be traced all the time. Second, the trajectories may be short. For example, Hoh et al. [35] collected a dataset containing GPS traces from 233 vehicles driving in a large US city and its suburban area. Each GPS sample does not contain any identification of drivers. Their data analysis shows a large number of very short trips, for example 30% of trips are shorter than 10 minutes, 50% of trips shorter than 18 minutes. This enables the attacker to break the privacy: by following a trace for only 10 minutes, the adversary may be able to track a vehicle from its home to a sensitive destination. Another possible privacy leakage is enabled when the attacker knows a series of locations that a specific user has been to (e.g., the attacker knows user Alice went to a beauty salon and a pharmacy). Then by matching the traces with the adversary knowledge, the attacker may be able to re-identify the drivers and thus know all the locations they have been to [57].

RFID-based Moving Objects. Due to huge advances in positioning technologies such as RFID, the data collected from RFID devices has become largely available. Unlike GPS traces, RFID trajectories need not be continuous [45]. However, it still can lead to privacy breaches. Terrovitis *et al.* [56] give an example of a company in Hong Kong called Octopus that collects daily trajectory data of Hong Kong residents who use Octopus smart RFID card. When a card holder A uses her Octopus card to pay at different convenience stores that belong to the same chain (e.g., 7-Eleven), her transaction records gave away a portion of her own trajectory. If this portion of trajectory uniquely identifies A, then by matching it with the published trajectory database, even though the IDs of users have be removed, A still can be re-identified, together with her unknown trajectory portions.

Location Based Services. The advancement of mobile devices and networking techniques have spurred new types of services such as online user navigation to avoid traffic jams and/or bad weather conditions, way finding, store finding and friend finding, as well as mobile commerce and surveying. Users can use social networking applications (e.g., Google Latitude) to share information about their geospatial context on the go. These services are called *location based services (LBS).*

There has been much work on privacy issues regarding the use of LBSs by mobile users. Most works define the privacy risk as linking of requests for services and locations to specific mobile users. Works in [18; 32] de-identify a given request or a location by using perturbation and obfuscation techniques. Anonymization based privacy protection is used in [22; 11; 32; 33]. Among these works, [33] focuses on the protection of links between users and sensitive locations, [22; 32] unlinks individual location points belonging to a user, and [11] enforces location points referring to the same set of users to be anonymized together always. [26] replaces the exact location of a user U with a so-called anonymizing spatial region (ASR) that contains at least $K-1$ other users. Furthermore, it defines a *reciprocity* privacy model which requires that a set of K users always be grouped together for a given K. As anonymization may ruin data utility, [25] proposes to use private location-dependent queries based on the theory of Private Information Retrieval (PIR), instead of using an anonymizer. [51] defines the notion of strong location privacy, which renders a query indistinguishable from any location in the data space. The strong location privacy is achieved by employing secure hardware-aided PIR. While these works focus on *on-line*, *service-centric* anonymity, we consider *off-line* and *data-centric* anonymity, as in the context of data publishing.

We refer the reader to the survey [30] on on-line service-centric data anonymity in location based services, and to [12; 23] for discussion on the relationship between trajectory anonymity in off-line static context and in on-line LBS context.

3. METHODS USING LOCATION BASED ADVERSARY KNOWLEDGE

In this section, we review recent papers [2; 45; 50; 56; 57] that consider location-based adversary knowledge. We will discuss the definition of location based adversary knowledge, the privacy model, and the anonymization techniques of these papers.

3.1 Adversary Knowledge and Privacy Model

In the context of publishing relational databases, there is a fundamental notion of *quasi-identifier* attributes, which correspond to the attributes (e.g., age, gender, and zipcode) that constitute public knowledge and that may be used as key for a linking attack leading to re-identification of the individuals [52; 53; 55]. The same reasoning exists for trajectory databases; the attacker may realize the existence of a set of quasi-identifier (QID) (location, time) pairs that may uniquely identify moving objects [11; 45; 56; 57]. In this context, QIDs are defined as (sets of) pairs of locations and timestamps.

Defining realistic quasi-identifiers is often challenging. It is not clear from where and how the knowledge of quasi-identifiers for each single user should be obtained. Both [11] and [57] argue that the quasi-identifiers may be provided directly by the users when they subscribe to the service, or be part of the users' personalized settings, or they may be found by means of statistical data analysis or data mining. However, defining spatio-temporal quasi-identifier in the real-world is not an easy task.

Given the aforementioned challenges, some works anonymize the trajectories without considering the QIDs of trajectories [2; 3; 50; 45], i.e., they are oblivious to the possible existence of QIDs. These three pieces of work are distinct from each other in that [2; 3] and [50] anonymize trajectories as a whole, assuming there does not exist any QID, while [45] anonymizes trajectories partially, without considering any specific QID but requiring that every sub-sequence of the trajectories of length L has to be shared by at least a certain number of moving objects.

We categorize the six papers [2; 3; 45; 50; 56; 57] that we will review in this section into two types: (1) QID-aware anonymization, and (2) QID-blind anonymization. QID-blind techniques do not consider any specific quasi-identifier when anonymizing the trajectories, while QID-aware techniques anonymize trajectories based on their QIDs.

t_{id}	trajectory
t_1	$a_1 \rightarrow b_1 \rightarrow a_2$
t_2	$a_1 \rightarrow b_1 \rightarrow a_2 \rightarrow b_3$
t_3	$a_1 \rightarrow b_2 \rightarrow a_2$
t_4	$a_1 \rightarrow a_2 \rightarrow b_2$
t_5	$a_1 \rightarrow a_3 \rightarrow b_1$
t_6	$a_3 \rightarrow b_1$
t_7	$a_3 \rightarrow b_2$
t_8	$a_3 \rightarrow b_2 \rightarrow b_3$

(a)

t_{id}	trajectory
t_1^A	$a_1 \rightarrow a_2$
t_2^A	$a_1 \rightarrow a_2$
t_3^A	$a_1 \rightarrow a_2$
t_4^A	$a_1 \rightarrow a_2$
t_5^A	$a_1 \rightarrow a_3$
t_6^A	a_3
t_7^A	a_3
t_8^A	a_3

(b)

Table 2: (a) an example trajectory database D, and (b) a local database D^A (A's knowledge).

3.1.1 QID-aware Anonymity Models

Terrovitis *et al.*'s QID definition and privacy model. Terrovitis *et al.* consider trajectories being simple sequences of addresses [56]. Let $\mathcal{P}$ be the domain of all addresses. Consider that $\mathcal{P}$ is partitioned into m disjoint non-empty sets of addresses $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_m$, where each adversary i controls a portion of addresses $\mathcal{P}_i$. For each trajectory t in the input database D, adversary i holds a portion (or a projection) t^i, thus adversary i holds a local database D^i containing the projections of all $t \in D$ with respect to $\mathcal{P}_i$. Table 2(a) shows an example of the original trajectory database D. The locations a_i (b_j) are controlled by adversary A (resp., B). Table 2(b) shows the projection D^A of the trajectory database that A controls.

A trajectory may appear multiple times in D and more than one trajectory may have the same projection with respect to $\mathcal{P}_i$. The most important property of a trajectory projection t^i is that adversary i can directly link it to the identities of all persons that pass through it, in its local database (e.g., loyalty program). Suppose D is an original trajectory database and D' is the transformed database that is published. For $t \in D$, let t^A be the projection of t on the locations controlled by adversary A. Suppose $p_j \notin t^A$ is a position not in the projection of a trajectory in the prtion controlled by A. Then what is the probability that A can associate (i.e., re-identify) to a real person? This is the measure of privacy that is of interest in [56] and is defined as

$$P(p_j, t^A, D') = \frac{|\{t' | t' \in S(t^A, D') \wedge p_j \in t'\}|}{|S(t^A, D')|},$$

where $S(t^A, D')$ is the set of trajectories in the published database D' which support t^A, i.e., each of them projects to t^A. Based on the quantified threats, Terrovitis et al. define the following privacy model.

Definition 3.1. [56] Given a database D of trajectories, where locations take values from P, construct a transformed database D', such that if D' is public, for all $t \in D$, every adversary A cannot correctly infer any location $\{p_j | p_j \in t \wedge p_j \notin t^A\}$ with probability larger than P_{br}, where P_{br} is a given breach probability threshold.

Yarovoy *et al.*'s QID definition and privacy model. Yarovoy *et al.* [57] argue that unlike in relational micro-data, where every tuple has the same set of quasi-identifier attributes, in mobility data we can not assume a set of particular locations, or a set of particular timestamps, to be a quasi-identifier for all the individuals. It is very likely that various moving objects have different quasi-identifiers and this should be taken into account in modeling adversarial knowledge. More precisely, given a moving object database $D = \{O_1, ..., O_n\}$ that corresponds to n individuals, and a set of m discrete time points $T = \{t_1, ..., t_m\}$, the QID is defined as a function:

$$QID : \{O_1, ..., O_n\} \rightarrow 2^{\{t_1,...,t_n\}}.$$

According to this definition, every moving object may potentially have a distinct quasi-identifier.

Based on the QID definition, the privacy model is defined as a notion of indistinguishability. More precisely, let D^* be a distorted (i.e., transformed) version of a trajectory database D. Two moving objects O, O' are indistinguishable in D^* at time t provided that $D^*(O, t) = D^*(O', t)$, i.e., both are assigned to the same region in D^*.

A straightforward way to define k-anonymity is as following: for every moving object O in D, there exist at least $k - 1$ other distinct moving objects $O_1, ..., O_{k-1}$, which are called the *anonymization group* of O, in D^*, such that $\forall t \in QID(O)$, O is indistinguishable from each of $O_1, ..., O_{k-1}$ at time t. Note that since different moving objects may have different QID, the anonymization groups associated with different objects may not be disjoint.

This is a fundamental shift from the situation for relational microdata and may lead to privacy breach of the plain k-anonymity model since it is possible that by combining overlapping anonymization groups, some moving objects may be uniquely identified. We will illustrate this with an example shortly. To capture such attack, Yarovoy *et al.* [57] define an *attack graph* associated with a trajectory database D and its distorted version D^*, as the bipartite graph G consisting of nodes for every individual I in D (called *I-nodes*) and nodes for every moving object id O (called *O-nodes*) in the published database D^*. G contains an edge (I, O) iff for each $t \in QID(I)$, $D(O, t) \sqsubseteq D^*(O, t)$.

An assignment of individuals to moving objects is *consistent* provided there exists a perfect matching in the bipartite graph G. Consider the trajectories in Figure 1(a) and its distorted database shown in Figure 1(b). Let $k = 2$ and $QID(O_1) = \{t_1\}$, $QID(O_2) = QID(O_3) = \{t_2\}$. Intuitively the best (w.r.t. information loss) anonymization group for O_1, O_2, and O_3 (with regard to their QIDs) are $AS(O_1) = \{O_1, O_2\}, AS(O_2) = AS(O_3) = \{O_2, O_3\}$. Clearly, the anonymization groups of O_1 and O_2 overlap. The corresponding attack graph is shown in Figure 1 (c). It is obvious that the edge (I_1, O_1) must be a part of every perfect matching. Thus, by constructing the attack graph an attacker may easily conclude that MOB O_1 can be re-identified as I_1.

Given the considerations above, Yarovoy *et al.* [57] formalize the attack model as follows. The attacker first constructs an attack graph associated with the published distorted version of D and the known QIDs as described above. Then, he repeats the following operation until there is no change to the graph:

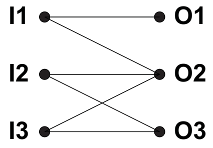
1. Identify an edge e that cannot be part of any perfect matching.
2. Prune the edge e.

MOB	t_1	t_2
O_1	$(1,2)$	$(5,3)$
O_2	$(2,3)$	$(2,7)$
O_3	$(6,6)$	$(3,6)$

(a)

MOB	t_1	t_2
O_1	$[(1,2),(2,3)]$	$(5,3)$
O_2	$[(1,2),(2,3)]$	$[(2,6),(3,7)]$
O_3	$(6,6)$	$[(2,6),(3,7)]$

(b)



(c)

Figure 1: Assuming $QID(O_1) = \{t_1\}$, $QID(O_2) = QID(O_3) = \{t_2\}$: (a) original database; (b) a 2-anonymity scheme that is not safe, and (c) the attack graph.

Next, he identifies every node O with degree 1. He concludes the (only) edge incident on every such node must be part of every perfect matching. There is a privacy breach if the attacker succeeds in identifying at least one edge that must be part of every perfect matching. We henceforth assume that unless otherwise mentioned, an attack graph refers to the graph obtained at the end of the above operations.

To defend against such attack, Yarovoy *et al.* [57] define the k-anonymity model as follows.

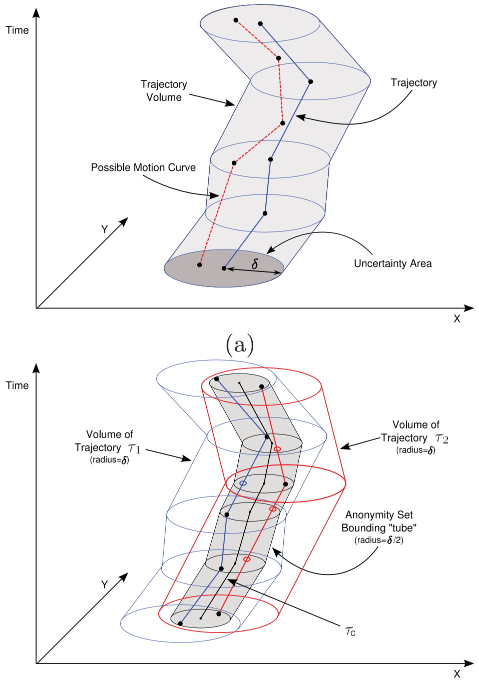
DEFINITION 3.2. [57] Let D be a trajectory database and D^* its distorted version. Let G be the attack graph w.r.t. D, D^*. Then D^* is k-anonymous provided that (i) every I-node in G has degree k or more; and (ii) G is symmetric, i.e., whenever G contains an edge (I_i, O_j), it also contains the edge (I_j, O_i).

An immediate observation is that in an attack graph that satisfies the above conditions, every O-node will have degree k or more as well.

3.1.2 QID-blind Anonymity Models

Abul et al.'s anonymity model [2; 3]. Abul *et al.* propose a novel concept of k-anonymity based on co-localization that exploits the inherent uncertainty of the moving object's whereabouts. Based on the observation that due to the imprecision in sampling and positioning systems (e.g., GPS), the trajectory of a moving object is represented a cylindrical volume instead of a polyline in a three-dimensional space. The position of a moving object in the cylinder then becomes uncertain. A graphical representation of an uncertain trajectory is shown in Figure 2.

The location uncertainty is captured by a radius parameter δ of the cylinder. Clearly all trajectories that move within the same cylinder are indistinguishable from each other. Based on this, Abul *et al.* [2] defined a (k, δ)-anonymity model as follows.


Time
Trajectory Volume
Trajectory
Possible Motion Curve
Y
δ
Uncertainty Area
X
Time
Volume of Trajectory τ_1 (radius=δ)
Volume of Trajectory τ_2 (radius=δ)
Anonymity Set Bounding "tube" (radius=δ/2)
Y
τ_c
X

(a)

(b)

Figure 2: (a) an uncertain trajectory: uncertainty area, trajectory volume and possible motion curve. (b) an anonymity set formed by two co-localized trajectories, their respective uncertainty volumes, and the central cylindrical volume of radius $\delta/2$ that contains both trajectories.

Definition 3.3. Given an anonymity threshold k and a radius parameter δ, a (k, δ)-*anonymity set* is a set of at least k trajectories that are co-localized w.r.t. δ.

They show that a set of trajectories S, with $|S| \geq k$, is a (k, δ)-anonymity set if and only if there exists a trajectory τ_c such that all the trajectories in S are possible motion curves of τ_c within an uncertainty radius of $\delta/2$. Given a (k, δ)-anonymity set S, the trajectory τ_c is obtained by taking, for each $t \in [t_1, t_n]$, the point (x, y) that is the center of the minimum bounding circle of all the points at time t of all trajectories in S.

Therefore, an anonymity set of trajectories can be bounded by a cylindrical volume of radius $\delta/2$. In Figure 2(b), we graphically represent this property.

The (k, δ)-anonymity framework requires to transform a trajectory database D to D^* such that for each trajectory $\tau \in D^*$ it exists a (k, δ)-anonymity set $S \subseteq D^*$, $\tau \in S$, and the distortion between D and D^* is minimized.

Nergiz et al.'s anonymity model [50]. Inspired by the *condensation* approach [4; 5], Nergiz *et al.* [50] consider the trajectories as a collection of points, each point represented by intervals on the three dimensions: $[x_1, x_2]$, $[y_1, y_2]$, and $[t_1, t_2]$. Based on this, they define the k-anonymity model as following:

Definition 3.4. *[50]* A trajectory database D^* is a k-anonymization of a trajectory dataset D if

- for every trajectory in D^*, there are at least $k-1$ other trajectories with exactly the same set of points;
- there is a one to one relation between the trajectories $tr \in D$ and trajectories $tr^* \in D^*$ such that for each point $p_i \in tr^*$ there is a unique $p_j \in tr$ such that $x_i^1 \leq x_j^1$, $x_i^2 \geq x_j^2$, $y_i^1 \leq y_j^1$, $y_i^2 \geq y_j^2$, $t_i^1 \leq t_j^1$, and $t_i^2 \geq t_j^2$, .

Given a set of trajectories that are going to be anonymized together, the key is to construct anonymity sets by creating point matching. Figure 3 shows an example of producing a point matching of three trajectories tr_1, tr_2, and tr_3. Unmatched points are suppressed in the anonymization.

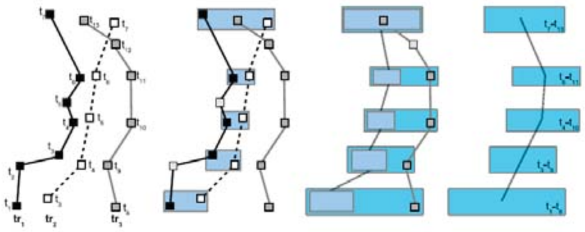


Figure 3: Anonymization of three trajectories tr_1, tr_2 and tr_3 via point matching

3.2 Anonymization Techniques

Next, we discuss the details of these anonymization techniques developed to achieve the anonymity models discussed above.

3.2.1 QID-aware Anonymization Techniques

Terrovitis *et al.*'s anonymization algorithm [56]. The main idea of Terrovitis *et al.*'s anonymization algorithm is to suppress the existence of certain points in the trajectories, taking into account the benefit in terms of privacy and the deviation from the main direction of the trajectory. As finding the optimal set of points to delete from trajectories with the minimum possible information loss is an NP-hard problem, the authors propose a greedy heuristic that iteratively suppresses locations, until the privacy constraint is met. The idea is that the algorithm simulates the attack from any possible adversary, and then resolves the identified privacy breaches.

First, the algorithm extracts the projected database D_i of each attacker i, according to his adversary knowledge (i.e., subsequences of each trajectory). Then the algorithm computes $sup(p_j, t^i, D)$ for each $p_j \in t, pj \notin t^i$. If there exists any unsafe projection, i.e., there exist pairs (P_j, t^i) such that $P(p_j, t^i, D) = \frac{sup(p_j, t^i, D)}{S(t^i, D)} > P_{br}$, where P_{br} is a user-defined privacy breach threshold, then the algorithm will unify a pair of projections (t_x^i, t_y^i), at least one of which is unsafe. Two projections can be unified only if one is a subtrajectory of the other, i.e., the larger projection contains all the points of the smaller one, and in the right order. For example, $a_1 \rightarrow a_3$ can be unified with a_3 (see t_5^A and t_6^A in Table 2(b)).

The reasoning behind unifying is that if t_x^i is an unsafe projection then either t_x^i is not supported in the transformed database D' resulting from this unification, or

$P(p_j, t_x^i, T') \leq P_{br}$ for all $p_j \notin P_i$. In the example of Table 2, after the unification of $a_1 \rightarrow a_3$ with a_3, trajectory t_5 becomes $t'_5 = a_3 \rightarrow b_1$ and the problems of both $a_1 \rightarrow a_3$ and a_3 are resolved; $a_1 \rightarrow a_3$ is no longer supported in D' and a_3 does not map to any B-location with probability higher than 50%.

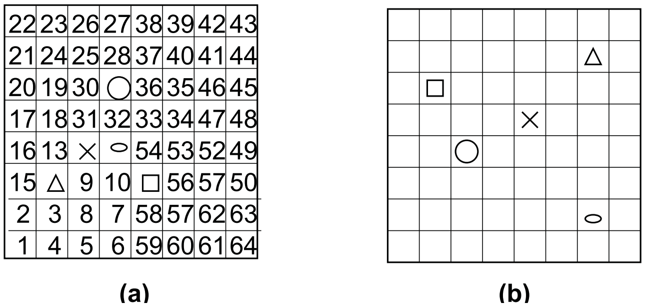
Since there may be more than one unsafe projection, the algorithm always picks the one of the lowest cost with respect to the information loss it entails. This cost is quantized by summing the distances of the transformed trajectories to the corresponding original ones, had the unification been committed, minus the corresponding cost before the unification. To improve the efficiency of the anonymization algorithm, the authors perform multiple unifications at each loop in the algorithm.

The experimental results of [56] were from synthetic trajectories of moving objects generated by using Brinkhoffs generator [14]. The results show that the anonymization algorithm performs the best if the number of unifications per iteration is large, the database size is large, and the location points in the adversary knowledge are distributed to many adversaries.

Yarovoy *et al.*'s anonymization algorithm [57]. Yarovoy *et al.* [57] study the case that QIDs of various moving objects may not be identical. Due to this fact, the design of anonymization algorithm is challenging since anonymization groups may not be disjoint, which is dramatically different from traditional anonymization techniques on relational databases where anonymization groups of different objects never overlap. Overlapping anonymization groups will result in revisits of earlier generalizations and possible re-generalization of existing anonymization groups with other objects, which will lead to significant computational complexity.

Yarovoy *et al.* proposed two different anonymization algorithms. Both make use of space filling curves for fast retrieval of nearest neighbors at every time point. Specifically, they use the Hilbert index of spatial objects for efficient indexing of moving objects at each time point. The Hilbert curve [34] is a continuous fractal space-filling curve that naturally maps a multi-dimensional space to one dimension. The Hilbert index of a list of points is assigned following the order in which the Hilbert curve visits these points in an n-dimensional space. It is well known that the Hilbert index preserves *locality*, i.e., points close in the multi-dimensional space remain close in the linear Hilbert ordering. To make use of this property, at each time point t, the Hilbert index of all moving objects will be constructed according to their locations at t. Figure 4 shows an example of the Hilbert index of locations.

To find the moving objects of the smallest aggregate distance from a moving object O over a set of time points, a local score for its Hilbert distance to the target moving object at the same timepoint is defined. The global score is equal to the sum of all local scores. The problem of finding moving objects with the top-$(k-1)$ closest aggregate distance from O thus reduces to finding the top-$(k-1)$ moving objects with the lowest overall score. Yarovoy *et al.* proposed the anonymization algorithm *GenAG* , which adopts a recent improvement of the well-known *Threshold Algorithm* (TA) [19; 20] and its variants (see [8]). In particular, the moving objects are stored in increasing order of the Hilbert index, and the TA algorithm is applied on the sorted Hilbert lists. As shown in [20], the TA algorithm has the *instance optimality* property. This property is inherited by *GenAG*. Intuitively, it guarantees that any algorithm, which does not make any wild guesses, cannot compute the answer to the top-k query (i.e., compute top-k nearest neighbors) in fewer accesses than the GenAG algorithm by more than a constant factor, on any input database.



22	23	26	27	38	39	42	43
21	24	25	28	37	40	41	44
20	19	30	○	36	35	46	45
17	18	31	32	33	34	47	48
16	13	×	⬭	54	53	52	49
15	△	9	10	□	56	57	50
2	3	8	7	58	57	62	63
1	4	5	6	59	60	61	64

						△	
	□						
				×			
		○					
						⬭	

Figure 4: Illustration of locations and their Hilbert indexes: (a) time t_1; (b) time t_2.

The basic idea of the anonymization algorithm *GenAG* is as following. First, for every time point t, we compute the Hilbert index $H_t(O)$ of every moving object O and insert the entries $(O, H_t(O))$ into a list L_t in increasing order of $H_t(O)$. These lists will be used repeatedly for computing the anonymization group of different subjects, where subject is the moving object w.r.t. which we may need to compute nearest neighbors. A key condition in the definition of k-anonymity is that the induced attack graph must be symmetric. To satisfy this requirement, Yarovoy *et al.* first proposed the Extreme Union (*EU*) method that can achieve the symmetry requirement at the expense of generalizing all objects in an anonymization group with regard to the QIDs of all moving objects in the group. Specifically, they take the union of the QIDs of all moving objects in the anonymization group of the object O and generalize all of them with regard to every time point in this union.

While *EU* does produce generalizations that are k-anonymous, it can result in considerable information loss. To generalize objects less aggressively than *EU* and still meet the k-anonymity condition, Yarovoy *et al.* then proposed the *Symmetric Anonymization* (SA) method. Instead of keeping the set of objects being generalized together fixed (as *EU* does), SA keeps the timestamps with regard to which a set of moving objects is generalized together, fixed and equal to the QID of the target object O_i.

After the anonymization groups are constructed, all moving objects in the same anonymization group will be generalized identically. However, due to the fact that QIDs may overlap, anonymization groups associated with different objects may not be disjoint, which will result in revisits of earlier generalizations and possible re-generalization of existing anonymization groups with other objects. The problem is to avoid the backtracking of generalization, so that the generalized position of every object at every timestamp can be computed in one shot. To achieve this goal, each moving object O_i with $t \in QID(O_i)$ is run through and $AG(O_i)$, the anonymization group, is added to its generalization set EC_t.

EC_t is maintained by using the well-known UNION/FIND data structure for disjoint sets, with path compression [21]. The algorithm has an amortized complexity of $O(n\alpha(n))$ for performing n operations consisting of union (i.e., merge) of two disjoint sets or finding the unique set to which an element belongs, where α is the inverse of the extremely fast growing Ackermann's function.

After the computation of generalization sets, a k-anonymization D^* of D can be obtained as follows. For each time t in the union of all QIDs, for each generalization set C w.r.t. time t, we generalize the position of every moving object $O \in C$ to the least upper bound of the positions of objects in C w.r.t. the lattice of rectangles.[5] This corresponds to the smallest region containing the positions of all moving objects in C at time t.

Yarovoy *et al.* [57] use two datasets for their experiments, a real-world trajectory dataset that is obtained by collecting traces of GPS-equipped cars moving in the city of Milan (Italy), and a synthetic dataset generated by using Brinkhoffs generator [14]. The observation is that the maximal size of the QIDs has a significant impact on the algorithm's performance. While run-time grows sub-linearly with k, it grows super-linearly with the maximal size of the QIDs.

3.2.2 QID-blind Anonymization Techniques

Next, we review the details of these papers.

Abul *et al.*'s anonymization algorithms [2; 3].

Abul *et al.* propose a two-step (k, δ)-anonymity anonymization method, called $\mathcal{NWA}$ ($\mathcal{N}$ever $\mathcal{W}$alk $\mathcal{A}$lone), that can efficiently anonymize a trajectory database with low information loss, by means of clustering and perturbation. In particular, as perturbation method is chosen *space translation*: i.e., slightly moving some observations in space. A suitable measure of the information distortion introduced by space translation is defined, and the problem of achieving (k, δ)-anonymity by space translation with minimum distortion is proven to be NP-hard.

In the first clustering step, the moving objects database D is partitioned in groups of trajectories, each group having size in the interval $[k, 2k-1]$. After having tried a large variety of clustering methods for trajectories under the k-member constraint, Abul *et al.* chose a simple greedy method as the best trade-off between efficiency and quality of the results. The method is further enhanced with ad-hoc preprocessing and outlier removal. In fact it is claimed by the authors (but also by other previous work, e.g., [15]), that outlier detection and removal might be a very important technique in clustering-based anonymization schemes: the overall quality of the anonymized database can benefit by the removal of few outlying trajectories.

The pre-processing step aims at partitioning the input database into larger equivalence classes w.r.t. time span, i.e. groups containing all the trajectories that have the same starting time and the same ending time. This is needed because $\mathcal{NWA}$ adopts Euclidean distance that can only be defined among trajectories having the same time span: if performed directly on the raw input data this often produces a large number of very small equivalence classes, possibly leading to very low quality anonymization. To overcome this problem, a simple pre-processing method is developed. The method enforces larger equivalence classes at the price of a small information loss. The pre-processing is driven by an integer parameter π: only one timestamp every π can be the starting or ending point of a trajectory. For instance, if the original data was sampled at a frequency of one minute, and $\pi = 60$, all trajectories are pre-processed in such a way that they all start and end at full hours. To do that, the first and the last suitable timestamps occurring in each trajectory are detected, and then all the points of the trajectory that do not lay between them are removed.

The greedy clustering method iteratively selects a pivot trajectory and makes a cluster out of it and of its $k-1$ unvisited nearest neighbors, starting from a random pivot and choosing next ones as the farthest unvisited trajectories w.r.t. previous pivots. Being simple and extremely efficient, the greedy algorithm allows to iteratively repeat it until clusters satisfying some criteria of compactness are built.

More in details, a compactness constraint is added to the greedy clustering method briefly described above: clusters to be formed must have a radius not larger than a given threshold. When a cluster cannot be created around a new pivot without violating the compactness constraint, the latter is simply *deactivated* — i.e., it will not be used as pivot but, in case, it can be used in the future as member of some other cluster — and the process goes on with the next pivot. When a remaining object cannot be added to any cluster without violating the compactness constraint, it is considered an outlier and it is trashed. This process might lead to solutions with a too large trash, in which case the whole procedure is restarted from scratch relaxing the compactness constraint, reiterating the operation till a clustering with sufficiently small trash is obtained. At the end, the set of clusters obtained is returned as output, thus implicitly discarding the trashed trajectories.

In the second step, each cluster of trajectories is perturbed by means of the minimum spatial translation needed to push all the trajectories within a common uncertainty cylinder, i.e., transforming them in an anonymity set.

Starting from a discussion on the limits of $\mathcal{NWA}$ Abul *et al.*, in a subsequent paper [3], we develop a novel method that, being based on EDR (Edit distance on Real sequences) [16] (instead of the Euclidean distance as it was $\mathcal{NWA}$), it has the important feature of being *time-tolerant*. The novel method is named $\mathcal{W}4\mathcal{M}$ ($\mathcal{W}ait\ for\ \mathcal{M}e$).

Another idea introduced in this follow-up paper, is to exploit the EDR computation also as a guide on how to perform the last step of the anonymization process. After having clustered trajectories it is needed to modify each cluster to make it an anonymity set. Being an edit distance, it is EDR itself to suggest how to do this *spatio-temporal editing*: this means that the computation done during the clustering phase can be reused in the points translation phase.

The experiments on both real and synthetic datasets confirm that $\mathcal{W}4\mathcal{M}$ produces higher quality (k, δ)-anonymized data than $\mathcal{NWA}$. However, it might be prohibitively expensive for large and complex datasets. Thus, Abul *et al.* develop techniques to make $\mathcal{W}4\mathcal{M}$ scalable. In particular, they introduce a novel $O(n)$ spatio-temporal distance function, named *LSTD* (linear spatio-temporal distance). Being linear, LSTD has the same computational cost of Euclidean distance, but it has not the same limits: in fact LSTD is time-tolerant, can be applied to trajectories of dif-

[5] The lattice of rectangles is defined using a discrete grid and the partial order is naturally based on containment.

ferent length, and it is tolerant to outliers. In practice, it represents a good trade-off between Euclidean distance and EDR.

Nergiz *et al.*'s anonymization algorithms [50]. As we discussed before, the key operation of Nergiz *et al.*'s technique is point matching. After defining and computing point matching for anonymization sets, the next challenge is to find the optimal anonymization of two trajectories. Nergiz et al. observed that there exists similarity between the problem of optimal trajectory anonymization and the string alignment problem in the context of DNA comparisons, where the goal is to find an alignment of strings such that total pairwise edit distance between the strings is minimized. By adapting the solution of string alignment problem, the alignment of two trajectories can be solved in polynomial time by using a dynamic programming approach.

Based on the definition of anonymization cost by using alignment of trajectories, Nergiz et al. proposed a greedy algorithm, called *multi TGA*, that is based on condensation based grouping algorithm. In particular, in each iteration, the algorithm creates an empty group G, randomly samples one trajectory $tr \in D$, puts tr into G, and initializes the group representative as $rep_G = tr$. Next, the closest trajectory $tr' \in TR \setminus G$ to rep_G is added to G, and then rep_G is updated. At the end of each iteration, a new group of k trajectories is formed. The iteration stops when there are less than k trajectories that remain ungrouped.

To reduce the significant cost of finding the closest trajectory to the group representative, a new algorithm (called *Fast TGA*) is introduced: in *Fast TGA* all the $k-1$ closest trajectories to the group representative are chosen in one pass. *Fast TGA* is faster by a factor of k but prodcues worse utility since it does not directly optimize against the log cost function. Indeed, as proven by Nergiz *et al.*, computing the optimal anonymization groups with minimal log cost for $n > 2$ trajectories in NP-hard. Therefore, Nergiz *et al.* adapted the heuristics of the string alignment problem to trajectory anonymization.

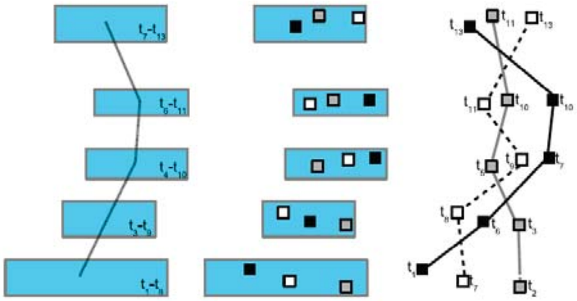


Figure 5: Example of reconstruction starting from the anonymization of Figure 3.

Nergiz *et al.* also discussed the drawbacks of applying the generalization-based anonymization techniques to trajectory databases. In particular, first, the generalized locations may disclose uncontrolled information about exact locations of the points. Second, the generalized trajectories may become useless for data mining and statistical applications which work on atomic trajectories. Therefore, Nergiz *et al.* adapt the reconstruction approach [4] and publish reconstructed data rather than data anonymized by means of generalization. An example reconstruction is shown in Figure 5. Compared with the generalization approach, the output after reconstruction is atomic and suitable for trajectory data mining applications.

Mohammed et al's anonymization technique [45]. Mohammed et al. define the *LKC-privacy* model that requires: (1) every sub-sequence with a maximum length L in the trajectory database has to be shared by at least a certain number of records, and (2) the ratio of sensitive value(s) in every anonymization group cannot be too high.

LKC-privacy is achieved by performing a sequence of suppressions on selected pairs from the trajectory database D. Mohammed *et al.* employ global suppression, meaning that if a pair p is chosen to be suppressed, all instances of p in D are suppressed. The purpose of using global suppression instead of local suppression is to retain exactly the same support counts of the preserved maximum frequency sequences in the anonymous trajectory database as there were in the raw data. The property of data truthfulness is vital in some data analysis tasks such as traffic analysis.

The LKC-privacy anonymization algorithm consists of two steps. The first step is to identify all sequences that violate the given LKC-privacy requirement. To avoid enumerating all possible violating sequences, the authors aim at finding minimal violating sequences. The solution is similar to Apriori algorithm [7]. It starts from size-1 sequences, i.e., all distinct pairs, as candidates. For each candidate, it scans D once to compute its frequency. If the sequence q violates the LKC-privacy requirement, then q is added to the MVS set, which stores minimal violating sequences; otherwise, q is added to the non-violating sequence set for generating the next candidate set.

The second step is a greedy algorithm that transforms the raw trajectory database D into an anonymous table D' with respect to a given LKC-privacy requirement by a sequence of suppressions. In each iteration, a pair p for suppression will be selected based on a greedy selection function. The greedy function, $Score(p)$, is to select a suppression on a pair p that maximizes the number of minimal violating sequences(MVSs) that have been removed but minimizes the number of maximal frequency sequences (MFS) removed in D. More formally, $Score(p)$ is defined as follows:

$$Score(p) = \frac{PrivGain(p)}{UtilityLoss(p) + 1}$$

where $PrivGain(p)$ and $UtilityLoss(p)$ are the number of minimal violating sequence (MVS) and the number of maximal frequent sequence (MFS) containing the pair p, respectively. Adding 1 to the denominator is to avoid dividing by zero when a pair p may not belong to any MFS, resulting in $|UtilityLoss(p)| = 0$.

The greedy algorithm assigns the initial $Score(p)$ to every candidate pair. In each iteration, the pair of the highest $Score(p)$ will be suppressed. After suppression, the score of the remaining candidate pairs will be updated. The iteration repeats until no candidate pair is available.

4. METHODS ON MOTION PATTERN BASED ADVERSARY KNOWLEDGE

In this section, we will review two papers [24; 37] that consider motion pattern based adversary knowledge. We will discuss the definition of the adversary knowledge, the privacy model, and the anonymization techniques of these papers.

4.1 Velocity-based Adversary Knowledge

Ghinita *et al.* [24] considered an attack based on static cloaking mechanisms, in which an adversary uses background knowledge of maximum speed to infer more specific location information. As an example, user Alice uses social networking applications (e.g., Google Latitude) to share information about her geo-spatial context, so that she can ask a nearby friend to join her for dinner, or to find on-going events close to her location. She reports her (cloaked) location as she moves. Assume that Alice has set her current on-line status to "Visiting shops in the down-town area". An attacker can infer with high probability that Alice is currently walking, hence her velocity can be no higher than 5 km/h. Alternatively, if Alice's status is "Out for a bicycle trip", her speed can be bound to at most 20 km/h. The attacker can estimate user Alice's possible region and intersect it with Alice's reported (cloaked) location. If a hospital building is situated in the intersected region, then the attacker can infer that Alice has a medical appointment, compromising her privacy.

Ghinita *et al.* considered two types of attacks: (1) the attacks without background knowledge about the sensitive locations on the map, and (2) the attacks with such background information. In the former case, the privacy requirement is not to allow an attacker to pinpoint the user location within a sub-region of a reported cloaked region. In the latter case, the privacy requirement dictates that the association probability between a user and a sensitive location must not exceed the user-specified threshold. Given a cloaked region A, the *probability of association* is formally defined as

$$\frac{\sum_{\forall f \in ft_i} Area(f \cap CR)}{Area(CR)}.$$

Based on these two cases, the privacy model is defined as

Definition 4.1. [24] Two cloaked regions A and B separated by time interval δt are safe to disclose in the attack model without background knowledge if $d_{haus}(A, B) \leq v\delta t$. Similarly, in the attack model with background knowledge the two regions are safe to disclose if $d_{pp}(A, B) = v\delta t$, where $d_{haus}(A, B)$ is the Hausdorff distance [10] between CRs A and B, and $d_{pp}(A, B)$ measures the maximum distance between any point in A to any point in B.

Ghinita *et al.* [24] consider two types of transformation on trajectory databases, temporal cloaking and spatial cloaking. Temporal cloaking is applicable when the partition of the map into cloaking regions (CRs) is fixed in advance. The authors propose two alternatives for achieving temporal cloaking: request deferral and postdating. In particular, consider user U who wants to issue a request at current time t_q . The location of u is enclosed as CR C. Previously at time t_A, u issued a request with CR A. Prior to entering C, u was situated inside region B, but no request with associated CR B was issued. At current time t_q , C is not safe to disclose, as it is too far away from A.

By *request deferral*, the request at t_q will be deferred until C becomes safe to disclose, i.e., until t_C s.t.

$$d(A, C) \leq v(t_C - t_A), t_C \geq t_A + d(A, C)/v,$$

where d can signify either the d_{haus} or d_{pp} distance. In this case, the request is delayed for a period of time equal to $t_C - t_q$.

By *postdating*, the request at t_q will be issued immediately, but using CR B. Since u is already outside B, the request will certainly incur some amount of space error. However, if the current position of u is not far away from B (e.g., u has only recently exited B), the error is likely to be low.

With the deferral and postdating primitives, the authors devise an heuristic that chooses the best of the two methods in order to maintain good QoS. The heuristics is based on the assumption that the user has the ability to predict (with reasonable accuracy) its position at a future time. This prediction will be used in evaluating whether deferral or postdating is more beneficial. Since the proposed solution is an heuristic in the first place, predicting future locations with high accuracy is not a necessity.

The second anonymization technique of the paper, spatial cloaking, is based on the assumption that the user's mobile device has sufficient resources, or when cloaking is performed by a trusted service, CRs can be dynamically computed at the time of the request. The challenge is to construct the CRs with the sets of sensitive features and associated sensitivity thresholds taken into considering. For example, assume that at some point along its trajectory, user u is situated inside a hospital H. Denote by $thr_H = 0.5$ the sensitivity threshold of u for feature type *hospital*. In this case, it is necessary to reduce the probability of association of u with H by creating a CR at least twice as large as the area of H. On the other hand, if the user is in a non-sensitive area, then the exact location could potentially be disclosed, since this is not a privacy violation.

The CR construction procedure consists of three steps. By the first step, features of the trajectories will be filtered; only sensitive features that intersect $MS(A)$, which encloses all locations where user u could be situated at request time t_q, will be kept to represent the set SF of candidates for inclusion in the CR. By the second step, the algorithm chooses a sensitive feature $f \in SF$ and enlarges it to find a CR that satisfies the privacy requirement, i.e., the sensitive area within the CR is of a fraction of the total CR area no larger than the user-specified threshold. By the third step, the algorithm checks whether the safety requirement is enforced and defer the request if necessary.

[24] used a set of trajectories that are randomly generated, and reported the space and time error by temporal cloaking and spatial cloaking on the dataset. The results show that at low velocity, it is likely that the requests are safe to be issued. As velocity increases initially, consecutive requests need to be deferred/postdated. However, as velocity continues to increase, the safety condition can be satisfied with only a short delay.

4.2 Correlation-based Adversary Knowledge

Jin et al. [37] pointed out that the locations of a user at continuous timestamps are indeed correlated. The attacker may exploit such fact and design a motion model to define the probability distribution of locations for a particular user, given the location of the user at the preceding h epochs (forward motion model), or the following h epochs (backward motion model). To be formal, a forward motion model is a conditional probability mass function

$$Pr[loc(p, t_j) = l_j | loc(p, t_{j-1} = l_{j-1}), \ldots, loc(p, t_{j-h} = l_{j-h})]$$

where $1 \le h \le j$ and $loc(p, t_j) = l_j$ indicates that the location of user P at time t_j is l_j. Similarly, a forward motion model is a conditional probability mass function

$$Pr[loc(p, t_j) = l_j | loc(p, t_{j+1} = l_{j+1}), \ldots, loc(p, t_{j+h} = l_{j+h})]$$

where $1 \le h \le j$. Both forward motion model and the backward motion models can be viewed as an h^{th}-order Markov chain. In their work, they mainly used a simple (forward and backward) linear motion model that is based on simple velocity distribution assumptions. Formally, the speed of each user is assumed to be uniformly distributed in the range $[v_1, v_2]$, and that the angle of motion is uniformly distributed in $[\theta_1, \theta_2]$. Given user p's location at timer t_1 as l_1, the possible locations for p at t_2 are as a sector with angles $[\theta_1, \theta_2]$ and distances $[r_1, r_2]$ where $r_1 = v_1 \times (t_2 - t_1)$, $r_2 = v_2 \times (t_2 - t_1)$.

Based on the forward and backward motion models, Jin *et al.* defined the privacy breach as following:

Definition 4.2. A release candidate $D^*(t_j)$ is said to cause a privacy breach if either of the following statements is true for user-defined breach threshold T:

$$max_{p,l_j} Pr[loc(p, t_j) = l_j | D(t_{j-1}), \ldots, D(t_{j-m}), D^*(t_j)] > T$$

$$max_{p,l_j} Pr[loc(p, t_j) = l_j | D(t_{j+1}), \ldots, D(t_{j+m}), D^*(t_j)] > T$$

The goal of publishing the trajectory database is to ensure that location trace data does not result in a privacy breach. First, Jin *et al.* proposed an pruning method to improve the performance of computation of both forward and backward privacy breach. Second, they designed a publishing protocol of the trajectories, suggesting increase the size or vary the composition of anonymization groups, and limit the frequency with which to publish a release candidate. They also proposed the concept of *durable* anonymization groups that contain the same pseudonyms at all epochs across time. In their empirical study, they analyzed the occurrence of the motion prediction inference problem, and evaluated the effectiveness of the publishing algorithms, including the pruning approach and the effect of using durable versus non-durable clusters. They implemented two protocols for data publishing. In the first protocol, the data is initially clustered into anonymization groups at epoch 1 using the clustering method proposed in [4], which is called $k-Condense$. This method takes as input a parameter k, and uses a heuristic to cluster the points into groups based on their proximity, such that each resulting group contains at least k points. With durable clusters, once the cluster is produced at the first epoch, the clusters are retained and simply checked at subsequent epochs for forward breaches. Data is published if the forward breach probability for each cluster is below the user-defined threshold. In the second protocol, the data is reclustered at each epoch, using the $k - Condense$ algorithm. At each epoch the breach probability is computed and the snapshot at an epoch is published if the forward and backward breach probability for each cluster is below the user-defined threshold.

Jin et al.[37] use real GPS traces from a study conducted by a Transportation Research Institute of University of Michigan, as well as synthetic trajectories. The experiments show that the correlation-based attack can be successful against both datasets. However, their approaches can be effective to defend against the attack.

5. CONCLUSION AND OPEN PROBLEMS

Location privacy has already been acknowledged as an important problem, and effective privacy-preserving solutions to publish trajectories will be necessary to support the widespread development and adoption of location-based applications. This survey discussed the state-of-the-art in anonymous personal mobility data publication, with the focus on the definition of adversary knowledge, the privacy model, and the anonymization algorithms.

There are several interesting open directions for future research in this area. We discussed earlier the challenge of deriving quasi-identifiers in the context of mobility data: as argued by some authors, they might be defined by the users themselves, or they might be "learnt" by mining the trajectory database. There are two fundamental issues here.

Granularities of QID Locations. With today's positioning systems, the locations of moving objects can be recorded very accurately, down to the level of (longitude, latitude) pairs. Indeed, this has led to some papers [2; 57] defining QIDs as a sequence of coordinates in the Euclidean space. However, in practice, a set of spatial *areas* as opposed to (long, lat) pairs, may be sufficient to identify moving objects with high probability. Examples of such spatial areas include landmarks such as living area, working buildings, and public places (e.g., an oncology clinic) [56; 40].

Defining a suitable granularity of QID locations is an important and challenging problem. There have been several approaches that define QIDs on various granularities. Besides coordinates in the Euclidean space [2; 57]), Kido *et al.* [38] divide the space into several regions; the attacker's adversary knowledge of position information is delimited by the region it belongs to. Monreale *et al.* [48] consider the *semantic trajectory* [54], which reasons over trajectories from a semantic point of view, and defines sensitive spatial areas and QIDs based on a "privacy places" taxonomy. Finding a realistic and actionable and computational definition of quasi-identifiers is an important open problem.

Efficient Discovery of QIDs. Several existing works (e.g., [11; 57]) assume that the QIDs can be provided either directly by the users when they subscribe to the location-based service or be part of the users' personalized settings. We argue that in practice, the quasi-identifiers are application dependent, and may not be known a priori. Therefore, it is necessary to develop algorithms that can efficiently compute QIDs from the trajectory databases.

Intuitively, QIDs are the sets of locations whose frequencies are nearly unique in the sense of potentially identifying an individual. Hence, the problem of finding QIDs is similar to the well-known problem of frequent pattern mining [6]. However, frequent pattern mining algorithms cannot be used directly to find QIDs, since frequent pattern mining returns the patterns whose frequency counts are *no less than* a given threshold, while QID mining looks for the patterns whose frequency counts are *no more than* a threshold. We cannot use infrequent pattern mining algorithms [17] either, as the proper subsets of infrequent patterns must be frequent, which does not hold for QIDs. How to efficiently discover QIDs from the trajectory databases, which are typically large scale, is an interesting problem that is worthy of further exploration.

A close and interesting research area is the so called *privacy-preserving data mining* [13], i.e., instead of anonymizing the data for privacy-aware data publication, the focus of privacy is shifted directly to the analysis methods. Few papers exist along this line of research. An example is [1] which addresses the problem of *hiding mobility patterns*, that is, we want to publish a database of trajectories of moving objects, in such a way that some sensitive patterns holding in the data can not be retrieved by means of pattern mining techniques. The authors show how a trivial solution may be not safe enough: in fact, in certain cases, a malicious adversary can exploit the background knowledge of the road network to reconstruct the original database.

Another line of research, not yet started, is about developing ad-hoc anonymization techniques for the intended use of the data: for instance, with respect to a specific spatio-temporal data mining analysis.

With the recent explosion of social applications over mobile devices, the activity of collecting and analyzing individual mobility data for the purpose of developing novel services is expected to have a growing importance in the next years. The privacy issues related to these activities pose technical problems for which only preliminary solutions exist, as we reported in this survey. Therefore there is plenty of interesting research opportunities and challenges in this rather young area.

6. REFERENCES

[1] Abul, O., Bonchi, F., and Giannotti, F. Hiding sequential and spatiotemporal patterns. *IEEE Trans. Knowl. Data Eng. 22*, 12 (2010), 1709–1723.

[2] Abul, O., Bonchi, F., and Nanni, M. $\mathcal{N}$ever $\mathcal{W}$alk $\mathcal{A}$lone: Uncertainty for anonymity in moving objects databases. In *Proc. of the 24nd IEEE Int. Conf. on Data Engineering (ICDE'08).*

[3] Abul, O., Bonchi, F., and Nanni, M. Anonymization of moving objects databases by clustering and perturbation. *Inf. Syst. 35*, 8 (2010), 884–910.

[4] Aggarwal, C. C., and Yu, P. S. A condensation approach to privacy preserving data mining. In *Proc. of the 9th Int. Conf. on Extending Database Technology, (EDBT'04).*

[5] Aggarwal, C. C., and Yu, P. S. On anonymization of string data. In *Proc. of the 2007 SIAM Int. Conf. on Data Mining.*

[6] Agrawal, R., Imieliński, T., and Swami, A. Mining association rules between sets of items in large databases. *SIGMOD Record 22* (June 1993), 207–216.

[7] Agrawal, R., and Srikant, R. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases* (1994), pp. 487–499.

[8] Akbarinia, R., Pacitti, E., and Valduriez, P. Best position algorithms for top-k queries. In *Proceedings of the 33rd international conference on Very large data bases* (2007), pp. 495–506.

[9] Ashbrook, D., and Starner, T. Using gps to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing 7*, 5 (2003), 275–286.

[10] Atallah, M. J. *Algorithms and Theory of Computation Handbook.* CRC Press, 1998.

[11] Bettini, C., Wang, X. S., and Jajodia, S. Protecting Privacy Against Location-Based Personal Identification. In *Proc. of the Second VLDB Workshop on Secure Data Management (SDM'05).*

[12] Bonchi, F. Privacy preserving publication of moving object data. In *Privacy in Location-Based Applications* (2009), C. Bettini, S. Jajodia, P. Samarati, and X. S. Wang, Eds., pp. 190–215.

[13] Bonchi, F., Saygin, Y., Verykios, V. S., Atzori, M., Gkoulalas-Divanis, A., Kaya, S. V., and Savas, E. Privacy in spatiotemporal data mining. In *Mobility, Data Mining and Privacy*, F. Giannotti and D. Pedreschi, Eds. Springer, 2008, pp. 297–333.

[14] Brinkhoff, T. Generating traffic data. *IEEE Data Eng. Bull. 26*, 2 (2003), 19–25.

[15] Byun, J.-W., Kamra, A., Bertino, E., and Li, N. Efficient k-anonymization using clustering techniques. In *Proc. of the 12th Int. Conf. Database Systems for Advanced Applications, (DASFAA'07).*

[16] Chen, L., Özsu, M. T., and Oria, V. Robust and fast similarity search for moving object trajectories. In *Proc. of the 2005 ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'05).*

[17] Dong, X., Zheng, Z., and Niu, Z. Mining infrequent itemsets based on multiple level minimum supports. *Second International Conference on Innovative Computing ,Information and Control, (ICICIC '07),.*

[18] Duckham, M., and Kulik, L. A Formal Model of Obfuscation and Negotiation for Location Privacy. In *Proc. of the Third Int. Conf. on Pervasive Computing (Pervasive 2005)* (2005), pp. 152–170.

[19] Fagin, R., Lotem, A., and Naor, M. Optimal aggregation algorithms for middleware. In *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems* (2001), pp. 102–113.

[20] Fagin, R., Lotem, A., and Naor, M. Optimal aggregation algorithms for middleware. *Journal of Computer System and Science 66* (June 2003), 614–656.

[21] Galil, Z., and Italiano, G. F. Data structures and algorithms for disjoint set union problems. *ACM Comput. Surv. 23* (September 1991), 319–344.

[22] Gedik, B., and Liu, L. Location Privacy in Mobile Systems: A Personalized Anonymization Model. In *Proc. of the 25th Int. Conf. on Distributed Computing Systems (ICDCS'05).*

[23] Ghinita, G. Private queries and trajectory anonymization: a dual perspective on location privacy. *Trans. Data Privacy 2* (April 2009), 3–19.

[24] Ghinita, G., Damiani, M. L., Silvestri, C., and Bertino, E. Preventing velocity-based linkage attacks in location-aware applications. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (2009), pp. 246–255.

[25] Ghinita, G., Kalnis, P., Khoshgozaran, A., Shahabi, C., and Tan, K.-L. Private queries in location based services: anonymizers are not necessary. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data* (2008), SIGMOD '08, pp. 121–132.

[26] Ghinita, G., Zhao, K., Papadias, D., and Kalnis, P. A reciprocal framework for spatial k-anonymity. *Information System 35* (May 2010), 299–314.

[27] gil Lee, J., and Han, J. Trajectory clustering: A partition-and-group framework. In *Proc. of the 2007 ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'07)* (2007), pp. 593–604.

[28] gil Lee, J., Han, J., and Li, X. Trajectory outlier detection: A partition-and-detect framework. In *Proc. of the 24th IEEE International Conference on Data Engineering (ICDE'08)* (2008).

[29] gil Lee, J., Han, J., Li, X., and Gonzalez, H. Traclass: Trajectory classification using hierarchical region-based and trajectory-based clustering ? abstract. In *Proc. of the 34th Int. Conf. on Very Large Databases (VLDB'08)* (2008).

[30] Gkoulalas-Divanis, A., Kalnis, P., and Verykios, V. S. Providing k-anonymity in location based services. *SIGKDD Explore Newsletter 12* (November 2010), 3–10.

[31] Gruteser, M., and Grunwald, D. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In *Proc. of the First Int. Conf. on Mobile Systems, Applications, and Services (MobiSys 2003).*

[32] Gruteser, M., and Hoh, B. On the Anonymity of Periodic Location Samples. In *Proc. of the Second Int. Conf. of Security in Pervasive Computing (SPC 2005)* (2005), pp. 179–192.

[33] Gruteser, M., and Liu, X. Protecting Privacy in Continuous Location-Tracking Applications. *IEEE Security & Privacy Magazine 2*, 2 (2004), 28–34.

[34] Hilbert, D. Über die stetige abbildung einer linie auf ein flächenstück. *Math. Ann. 38* (1891), 459–460.

[35] Hoh, B., Gruteser, M., Xiong, H., and Alrabady, A. Preserving privacy in gps traces via uncertainty-aware path cloaking. In *Proceedings of the 14th ACM conference on Computer and communications security* (2007), pp. 161–171.

[36] Jeung, H., Liu, Q., Shen, H. T., and Zhou, X. A hybrid prediction model for moving objects. In *Proc. of the 24th IEEE International Conference on Data Engineering (ICDE'08)* (2008).

[37] Jin, W., LeFevre, K., and Patel, J. M. An online framework for publishing privacy-sensitive location traces. In *Proceedings of the Ninth ACM International Workshop on Data Engineering for Wireless and Mobile Access* (2010).

[38] Kido, H., Yanagisawa, Y., and Satoh, T. Protection of Location Privacy using Dummies for Location-based Services. In *Proc. of the 21st IEEE Int. Conf. on Data Engineering (ICDE'05).*

[39] Kido, H., Yanagisawa, Y., and Satoh, T. An Anonymous Communication Technique using Dummies for Location-based Services. In *Proc. of the Third Int. Conf. on Pervasive Computing (Pervasive 2005)* (2005), pp. 88–97.

[40] Krishnamachari, B., Ghinita, G., and Kalnis, P. Privacy-preserving publication of user locations in the proximity of sensitive sites. In *Proceedings of the 20th international conference on Scientific and Statistical Database Management* (2008), SSDBM '08, pp. 95–113.

[41] Krumm, J. Inference attacks on location tracks. In *the Proceedings of the Fifth International Conference on Pervasive Computing (Pervasive)* (2007).

[42] Li, X., Han, J., Kim, S., and Gonzalez, H. Anomaly detection in moving object.

[43] Li, X., Han, J., Lee, J.-G., and Gonzalez, H. Traffic density-based discovery of hot routes in road networks.

[44] Mamoulis, N., Cao, H., Kollios, G., Hadjieleftheriou, M., Tao, Y., and Cheung, D. W. Mining, indexing, and querying historical spatiotemporal data.

[45] Mohammed, N., Fung, B. C., and Debbabi, M. Walking in the crowd: anonymizing trajectory data for pattern analysis. In *Proceeding of the 18th ACM conference on Information and knowledge management* (2009), pp. 1441–1444.

[46] Mokbel, M. F., Chow, C.-Y., and Aref, W. G. Casper: Query processing for location services without compromising privacy. In *Proceeding of the 32nd International Conference on Very Large Databases (VLDB'06).*

[47] Mokbel, M. F., Chow, C.-Y., and Aref, W. G. The new casper: A privacy-aware location-based database server. In *Proc. of the 23rd IEEE International Conference on Data Engineering (ICDE'07).*

[48] Monreale, A., Trasarti, R., Renso, C., Pedreschi, D., and Bogorny, V. Preserving privacy in semantic-rich trajectories of human mobility. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS* (2010), pp. 47–54.

[49] Nanni, M., and Pedreschi, D. Time-focused clustering of trajectories of moving objects. *Journal of Intelligent Information Systems 27*, 3 (2006), 267–289.

[50] Nergiz, E., Atzori, M., and Saygin, Y. Towards trajectory anonymization: a generalization-based approach. In *Proc. of ACM GIS Workshop on Security and Privacy in GIS and LBS* (2008).

[51] Papadopoulos, S., Bakiras, S., and Papadias, D. Nearest neighbor search with strong location privacy. *Proc. VLDB Endow. 3* (September 2010), 619–629.

[52] Samarati, P., and Sweeney, L. Generalizing data to provide anonymity when disclosing information (abstract). In *Proc. of the 17th ACM Symp. on Principles of Database Systems (PODS'98).*

[53] Samarati, P., and Sweeney, L. Protecting Privacy when Disclosing Information: k-Anonymity and its Enforcement Through Generalization and Suppresion. In *Proc. of the IEEE Symp. on Research in Security and Privacy* (1998), pp. 384–393.

[54] Spaccapietra, S., Parent, C., Damiani, M. L., de Macedo, J. A., Porto, F., and Vangenot, C. A conceptual view on trajectories. *Data Knowledge Engineering 65* (April 2008), 126–146.

[55] Sweeney, L. k-anonymity privacy protection using generalization and suppression. *International Journal on Uncertainty Fuzziness and Knowledge-based Systems 10*, 5 (2002).

[56] Terrovitis, M., and Mamoulis, N. Privacy preservation in the publication of trajectories. In *Proc. of the 9th Int. Conf. on Mobile Data Management (MDM'08).*

[57] Yarovoy, R., Bonchi, F., Lakshmanan, L. V. S., and Wang, W. H. Anonymizing moving objects: How to hide a MOB in a crowd? In *Proc. of the 12th Int. Conf. on Extending Database Technology (EDBT'09).*

[58] Zheng, Y., Zhang, L., Xie, X., and Ma, W.-Y. Mining interesting locations and travel sequences from gps trajectories. In *Proceedings of the 18th international conference on World wide web* (2009), pp. 791–800.