GEPETO: a GEoPrivacy-Enhancing TOolkit

Sébastien Gambs
IRISA/INRIA - Université de Rennes 1
Campus Universitaire de Beaulieu, 35042 Rennes, France
Email: sgambs@irisa.fr

Marc-Olivier Killijian and Miguel Núñez del Prado Cortez
CNRS ; LAAS ;
7 avenue du Colonel Roche, F-31077 Toulouse, France
Université de Toulouse ; UPS, INSA, INP, ISAE ; LAAS ;
F-31077 Toulouse, France
Email: Marco.Killijian@laas.fr and mnpc@computer.org

***Abstract*—A geolocalised system generally belongs to an individual and as such knowing its location reveals the location of its owner, which is a direct threat against his privacy. To protect the privacy of users, a sanitization process, which adds uncertainty to the data and removes some sensible information, can be performed but at the cost of a decrease of utility due to the quality degradation of the data. In this paper, we introduce GEPETO (for *GEoPrivacy-Enhancing TOolkit*), a flexible open source software which can be used to visualize, sanitize, perform inference attacks and measure the utility of a particular geolocalised dataset. The main objective of GEPETO is to enable a user to design, tune, experiment and evaluate various sanitization algorithms and inference attacks as well as visualizing the following results and evaluating the resulting trade-off between privacy and utility.**

I. Introduction

We can define a *geolocalised system* as an object or a device which has an associated location. It can be, for example, a smartphone or a GPS-equipped vehicle. Usually, a geolocalised system belongs to an individual (or to a group of individuals, such as a family) and, as such, its location matches the location of its owner(s). Among all the *Personal Identifiable Information* (PII), learning the location of an individual is one of the greatest threat against his privacy. For instance, the spatio-temporal data of an individual can be used to infer the location of his home and workplace, to trace his movements and habits, to learn information about his centers of interests or even to detect a change from his usual behaviour. Moreover, if an adversary has some auxiliary knowledge, he can use it in combination with the location information to gain additional knowledge. For instance, if the adversary has access to the social network of an individual, he can determine when the person is visiting a given friend.

When collecting the mobility traces of individuals for a particular purpose, simply removing the identifiers of these persons or replacing it by a pseudonym is usually not sufficient to protect their privacy. Instead, a *sanitization* process, which adds uncertainty to the data and removes some sensible information, has to be performed. This loss of data, incurred by the sanitization process, comes with a dilemna: it certainly brings some privacy guarantees but at the cost of a decrease of utility due to the quality degradation of the data. Therefore, there is often a trade-off between the utility of the global task and the privacy protection of individuals.

The main purpose of this paper is to introduce GEPETO (*GEoPrivacy-Enhancing TOolkit*), a flexible open source software for managing geolocalised data. GEPETO can be used to visualise, sanitize, perform inference attacks and measure the utility of a particular geolocalised dataset. For each one of these actions, a set of different techniques and algorithms can be applied. The global objective of GEPETO is to enable a user to design, tune, experiment, and evaluate various sanitization algorithms and inference attacks as well as visualizing the following results and assessing their utility. In spirit, GEPETO is very close to the aim of the GeoPKDD[1] project whose goal was to integrate in a unified approach the aspects of privacy and knowledge mining on geolocalised data [4].

The outline of the paper is the following. First, in Section II, we discuss some privacy issues raised by geolocalised systems, as well as reviewing some sanitization algorithms which can be used to protect the privacy of the spatio-temporal data of users of such systems. Afterwards, in Section III, we describe the design and architecture of GEPETO, as well as the functionnalities currently implemented. Finally, we report in Section IV on some preliminary results obtained on a public geolocalised dataset concerning taxi movements within the San Francisco Bay area before concluding in Section V.

II. Geolocalised Systems and Privacy

A. Geolocalised Data

A *geolocalised dataset* D is a dataset which contains mobility traces of individuals. Technically, this data may have been collected either by recording locally the movements of each geolocalised system for a certain period of time, or centrally by a server which can track the location of these systems in real-time. A *mobility trace* is characterized by:

- An *identifier*, which can be the real identifier of the device (e.g. "Alice's phone"), a pseudonym or even the value "unknown" (when full anonymity is desired). A pseudonym is generally used when we want to protect the true identity of the system while still being able to link different actions performed by the same user.
- A *spatial coordinate*, which can be a GPS position (e.g. latitude and longitude coordinates), a spatial area (e.g. the name of a neighbourhood in a particular city) or even a semantic label (e.g. "home" or "work").

[1] http://www.geopkdd.eu

- A *time stamp*, which can be the exact date and time or just an interval (e.g. between 9AM and 12AM).
- Additional information such as the speed and direction for a vehicle, the presence of other geolocalised systems or individuals in the direct vicinity or even the accuracy of the estimated reported position. For instance, some geolocalised system are able to estimate the precision of their estimated location as depending on the number of GPS satellites they are able to detect.

A *trail of traces* is a collection of mobility traces that corresponds to the movements of an individual over some period of time. A geolocalised dataset D is generally constituted by an ensemble of trail of traces for different individuals.

A special case of mobility traces is called *contact traces* and consists in the recording of encounters between different devices. This kind of trace is composed of the identifiers of the devices and a time stamp. It may be recorded for instance by a device which has no integrated capacity for geopositioning but is capable of probing his neighbourhood to detect the presence of other devices (e.g. using Bluetooth neighbor discovery). In this paper, we focus on preserving the privacy of individuals in the context of mobility traces. Although sanitizing contact traces is equally interesting, it requires different techniques that are out of the scope of this paper.

B. Inference Attacks

In this paper, we consider the model where an *adversary* is attempting to cause some privacy breaches about an individual whose movements is contained in a particular geolocalised dataset D. The adversary may have some *a priori* knowledge such as the presence (or not) of a particular individual within D, a partial knowledge of his attributes (for example the location of his home or work), a model of his habits, his social network, the distribution of attributes within the population, the geographical knowledge of the road, ... Used in combination with the geolocalised data, this *a priori* knowledge may help the adversary to infer some private information.

An *inference attack* is an algorithm that takes as input a geolocalised dataset D, possibly together with some auxiliary information aux, and produces as output some additional knowledge. For example, an inference attack may consist in identifying the house or the place of work of an individual. This attack can be implemented straightforwardly provided that the adversary has access to a *reverse geocoding* tool[2] which maps a GPS position to the label of the corresponding physical place (for instance the address or the name of the building). A simple heuristic to identify the house of a person is to consider his last stop before midnight whereas to find his place of work requires simply to look for a location with few movements during the day.

Hoh, Gruteser, Xiong and Alrabady have performed a study [6] on the geolocalised data of vehicles within the Detroit area (Michigan, USA). The goal of their study was to automatically discover the home of the vehicles' drivers. Finding the home of a person based on the trail of traces of his vehicle is of course harder than tracking his exact movements with the position of his cell phone. The authors have used the following inference attack to automatically identify the houses:

- Remove all samples of vehicles which are moving at a speed greater than one meter per second (this information was part of the data).
- Select an area of interest (for instance a particular neighbourhood) rather than the global map in order to reduce the computation cost of the method.
- Apply a clustering algorithm to this area which groups close locations and may correspond to the same vehicle/individual in the same cluster.
- Filter clusters where there is no trace with an arrival time during the evening or which are outside residential areas.
- Consider that the home location of an individual is located as the median point within a cluster.

Among the 2 neighbourhoods and the 65 persons on which the authors have focused, the estimated houses correspond at 85% to the houses that a human would have recognized[3].

In GEPETO we have first considered the inference attack which attempts to identify the house of an individual from his trail of traces by finding the locations where the GPS system is switched on/off. The simple idea behind this attack is that a geolocalised system (such as a car or a cell phone) is generally switch on/off when its user begins (or ends) his day. This attack is very efficient in terms of computational ressources because it only requires to follow a trail of traces of an individual until a large time window with no traces is detected (for instance for longer than 2 hours). The traces before and after this time window are considered as being the begin/end locations of a typical day for the user.

More advanced inferences attacks can be used to detect the places of interests of a particular user (for instance his work or favorite places). From his favorite places, one may infer his interests, such as his taste for movie or a particular sport. Moreover, if the adversary observes that two individuals are often in close proximity, he can infer a social link between the two or when one of them is visiting the other. Once this information is gathered, it becomes possible for an adversary to build a model of the behaviour of an individual and detect when he is deviating from his usual one. The inference process is generally an incremental one where the adversary augments his knowledge about users contained in the geolocalised dataset by performing different attacks successively.

C. Sanitization and Utility

A *sanitization algorithm* S takes as input a geolocalised dataset D, introduces some uncertainty and removes some information from this dataset. S produces as output D', a sanitized version of the original dataset D. The main idea

[2]Google Maps (http://maps.google.com/) for instance offers the possibility of entering a GPS coordinate directly into its standard interface and returns the name of the corresponding physical location.

[3]As the exact identity of the drivers have been kept secret it was not possible for the authors to compare directly the houses returned by the algorithm against the ground truth (i.e. the exact address of the drivers) which explained why this particular evaluation method was chosen.

behind sanitization is that, for a potential adversary, breaching the privacy of a particular user is harder when working on D' than with D. A sanitization procedure usually comes with some privacy guarantees. For instance it can guarantee that at each time step there is a given number of individuals with a similar profile in each region of space. Possible sanitization techniques include:

- *Pseudonymization* replaces the common identifier of several mobility traces by either a randomly generated pseudonym (thus providing anonymity but not unlinkability) or by the $unknown$ value (thus granting full anonymity and unlinkability)[4]. Pseudonymization is generally performed as the first step of a sanitization process but as such it is often not sufficient for protecting the privacy of individuals.
- A *sampling* mechanism summarizes several mobility traces of a given user into fewer traces, generally by compressing an ensemble of traces that have occured within some time window into one median or average trace. By decreasing the total number of traces, sampling has the additional benefit that it compresses the data and, henceforth, reduces the computational ressources needed to further sanitize the data.
- *Perturbation* methods [1] change the spatial coordinate of a mobility trace by adding some random perturbation. For example, this noise can either be generated uniformly or using Gaussian noise within a sphere of radius r centered on the original coordinate. If the geography of the surrounding area is not taken into account during the perturbation, it may happen that the generated coordinate corresponds to a location which has no physical sense (for instance in the middle of a river or on a cliff).
- *Aggregration* merges several mobility traces into a single spatial coordinate. For instance, this spatial coordinate can be a surrounding spatial area such as a neighbourhood or an average of the mobility traces. During data preprocessing, a *clustering algorithm* (such as k-means) can be used to group traces that are close together into the same cluster while putting traces that are significantly distant in different clusters. This can be used to detect which traces should be merged together in an *aggregation* step. Another possibility is to detect which traces are occupying the same spatial area (for instance the same neighbourhood) at a certain moment in time and to replace each one of these individual traces by the same coordinate.
- *Spatial cloaking* [5] is an extension of the concept of k-anonymity [9] to spatio-temporal data and a form of aggregation. The main idea is to ensure at each time step, each individual is located within a spatial area that is shared by a least $k-1$ other individuals. This spatial area is reported instead of the exact location of these individuals, thus guaranteeing that even if an adversary can target the group where an individual is located, his behaviour will be indistinguishable from at least $k-1$ other individuals (k is a privacy parameter of the algorithm). An approach to achieve the property of spatial cloaking is to split recursively the space into areas of different sizes, until each area contains at least k individuals.
- *Mix-zones* [2] are inspired from the concept of mix-nets due to Chaum used for the anonymous communication of messages inside a network [3]. Mix-zones are spatial areas where (1) no measurements about the locations of individuals are performed and (2) such that each individual entering a mix-zone will have a different pseudonym when he exits the mix-zone. The main purpose of a mix-zone is to make it more difficult to link the different actions of an individual. Place of work or buildings with a high trafic are usually good candidates for mix-zones.
- *Swapping* consists in exchanging the mobility traces of two different individuals/pseudonyms for a certain period of time. For instance, we could swap the mobility traces of Alice with the traces of Bob during one day to render the behaviour of Alice more atypical and less predictable.
- *Removing* the mobility traces that are deemed too sensible can also be considered as a sanitization procedure. In the same spirit, it is also possible to *add fake records* inside the geolocalised dataset D' to blend the true movements of individuals inside artificial data.

As sanitization leads to a loss of information, it is important to have a *utility metric* in order to compare the utility of the original dataset D and the sanitized one D'. The utility measure can either be generic, for instance it can be linked to some global statistical properties of the dataset, or application-dependent, in which case it evaluates how well a particular application can be performed by using D' instead of D.

III. Design and Implementation of GEPETO

The global objective of GEPETO is to provide researchers concerned with geoprivacy with means to evaluate various sanitization techniques and inference attacks on geolocalised data. GEPETO provides an interface for the management of geolocalised data and offers several ways to manipulate this data such as sampling mechanisms, sanitization algorithms, inference attacks and a visualisation tool to display this data on a world map. The main idea is to offer a generic and flexible tool so that anyone can easily plug a new sanitization technique or a smart inference algorithm to attack geoprivacy. Moreover, the utility and visualization components provide means to evaluate the benefits of sanitization with regard to the success of inference attacks.

A. GEPETO Design

GEPETO is designed following a multi-layer architecture with the intended goal of making the system functional,

[4]*Anonymity* can be defined as being able to perform a particular action without having to reveal his identity whereas *unlinkability* is a stronger notion that involves not being able to link two different actions that have been performed by the same user. Typically, performing different actions under a pseudonym (instead of using his real name) provides anonymity but not unlinkability. See [7] for more details.

efficient, scalable, easily modifiable and reliable. First, the data layer is a set of classes which manages the communication with the database server for inserting, updating and deleting geodata. A control layer is in charge of the presentation, the local management and control of the data and provides a model of the data. The application layer is where the utility functions, the inference attacks and sanitization techniques are implemented. Finally, the visualization layer constitutes the graphical user interface of GEPETO where the user can load data, apply algorithms and visualize the results.

This layered architecture is targeted to provide a good separation of concerns between data access and data presentation, so that it is easy to implement new algorithms in the application layer, access and visualize data using the services of the control and the presentation layers. In GEPETO, the presentation layer uses external web-services for the visualization of the data such as Google Maps or Yahoo Maps. The design choices behind this architecture imply both benefits and drawbacks: GEPETO cannot be used offline as it needs access to the database server as well as to the internet in order to visualize data; but the implementation and maintenance is handle more easily this way, with a clear separation between the database and the visualization parts.

B. GEPETO Implementation

GEPETO is an open source software implemented in Java (JSDK 6.0) to make it independent from the operating system and designed following an object-oriented methodology with an iterative approach during development. The final design includes 9 packages, with a total of 60 classes. Presently, we have implemented the following 5 sanitization techniques and 2 inference attacks :

- *k-means clustering* on a single trail of traces, that can be parameterized by k, the target number of clusters;
- *Breadth-first search clustering*, with the number of points per cluster being a parameter;
- *Downsampling*, with the time window in seconds as an input parameter;
- *Pseudonymization*, where a seed to the pseudo-random generator can be entered (so that an experiment can be repeated under the same conditions);
- *Random perturbation* (with Gaussian noise), the standard deviation for the perturbation and a seed for the pseudo-random generator are taken as input parameters;
- *Begin and end location finder*, parameterized by the duration of a break in seconds;
- *Timely position finder*, parameterized by the begin and end time.

GEPETO has been explicitely designed to be extendable and it is easy to add new classes which implement another sanitization or inference algorithm. Scalability is a feature which is highly dependant of the memory available to run the algorithms and to process the given load of data. Indeed, the larger the volume of data processed is, the larger the computational ressources need to be. Currently, we have worked within 1024Mb and 1536Mb of memory in order to run the implemented algorithms on a dataset with about 10 millions of mobility traces. The database which stores the geolocalised data was implemented in MySQL 5.0.37 .

The sanitization process is often done in several incremental steps starting from the original data and the applying a first sanitization algorithm A, storing the intermediate result, and then applying a second algorithm B on the resulting data. For instance, the user may first pseudonymize the data, then perform a downsampling, before perturbing it and clustering it. The visualisation part of GEPETO allows to get a clear picture of the data evolution from the original geolocalised data to a desired sanitized data.

IV. Preliminary Results

In this section, we illustrate the use of GEPETO in evaluating sanitization techniques, utility functions and inference attacks. For this purpose, we use a public geolocalised dataset taken from the CRAWDAD repository [8]. This dataset contains mobility traces of taxi cabs with the San Francisco bay area, USA. It contains GPS coordinates of approximately 500 taxis collected over 30 days in the San Francisco Bay Area. The dataset contains about 500 trails of data, each trail containing around 20000 mobility traces. Indeed, the GPS coordinates were recorded every minute during 30 days, with approximately 12 hours of recordings per day.

A. Inference Attacks

For the sake of demonstration, we begin here with illustrating how GEPETO can be easily used to infer some private data about the taxi cabs, such as their home address for example. At first, GEPETO can be used to simply visualize the various trails, and trying to caracterize the geolocalised data contained in the dataset. When visualizing the data on the San Francisco map, one can easily recognize some hotspots, such as the San Francisco International Airport or various train and taxi stations. These hotspots being places where the taxis usually wait for customers during some period of time, many traces are plots on these spots.

A second step is to say that the beginning and ending locations of the taxis, for each working day, might convey some meaningful information, such as their home or company address. However, taxis not only work during daytime and thus, finding the first trace for each day is not sufficient. This is the purpose of the *begin and end location finder* inference attack implemented in GEPETO. With this algorithm, one can say that a new period of work starts after a given break duration, say 2 hours. Thus the algorithm looks for such breaks, and extracts the trace before the break as the ending location, and the trace after the break, as the beginning location. We must say that this attack has been very fruitful.

A first interesting inference was the identification of the taxi company main parking location. Indeed, many cabs come back and forth from this location, as they park their cab at the company lot. We were able to verify this statement simply using the yellow pages of San Francisco.

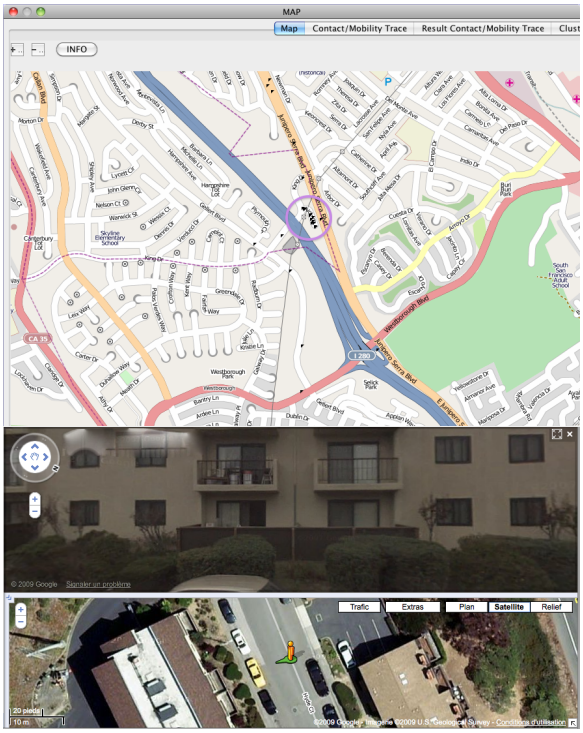

Fig. 1. A successful *begin and end location finder* inference attack

The second category of statements that could be inferred from this attack directly concerns private information of the individual taxi drivers[5]. During this study, we examined the trails of 90 individual taxis, chosen at random among the 135 first in the dataset. We used GEPETO to visualize the data of these 90 taxis after applying the *begin and end location finder* inference attack, manually picking those whose data seemed the most fragile. For 20 of these 90 taxis, the visualization of the resulting data show a narrow neighbourhood for their homes with a pretty high confidence. Please note that, as we do not have the real addresses of the taxis, we were unable to formally validate these statements. However, as shown in the remainder of this section, we were able to use Google Maps to validate some of the inferred data. Indeed, for 10 of the 90 taxis checked, the attack resulted in an address (or a small portion of a street) where the taxi was parked during most of the breaks. This address is most probably the home address of the taxi driver. In Figure 1, one can see the result of the attack, a Google Maps view and a StreetView of the address. For the remaining 70 taxis examined, the *begin and end location finder* inference attack simply identified hotspots.

B. Sanitization and Utility

In order to investigate how well GEPETO can protect privacy, we have tested several sanitization algorithms on different trails of traces of taxis. The impact of the sanitization process can be measured both by looking at the success of the inference attack on the sanitized data (and compare it with the result obtained on the non-sanitized one) and by evaluating how well some global property of the system is preserved. More precisely we have looked if sanitization methods, such as downsampling and random distortion, can conceal the home of taxi drivers against the inference attack described in the previous section and how these methods influence the average speed of these taxis deduced from the resulting traces.

In the CRAWDAD dataset, a taxi generates on average a mobility trace with its actual position and a time-stamp every minute. The effect of a downsampling is to summarize several traces contained within a time window of fixed length into one single trace. The basic downsampling method we have implemented takes the median trace of the time window[6] as the stored representative. One advantage of this variant of sampling is that the representative is always a location which is physically meaningfull (which is not necessarily the case if instead we set the average as the representative).

Regarding privacy, downsampling has the effect of hiding the exact departure point of a taxi as the sampled position is generally located a few minutes away from the departure position (which constitutes the beginning of the time window). For taxi drivers where it was easy to find the home adress from the original data, we observed that downsampling with a time window of length 300 and 450 seconds usually leaves 2 or 3 places spread around a relatively large area as potential candidates[7]. When we increase the length of the time window to 600 seconds and above, it becomes more and more difficult to target even a specific area as potential neighbourhood for the home of the taxi driver. Finally, going up to 3600 seconds (one hour) render this search almost impossible.

Concerning the utility as measured by the average speed of taxi, downsampling has the effect of decreasing significantly the speed reported. This can be easily explained by observing that the distance between two traces is approximated by a straight line which underestimate the true distance. Downsampling magnifies this effect by summarizing a set of relatively close traces located in the same window by a single one.

We have also tested the random distortion method centered on the initial location with the application of Gaussian noise. In practice, applying a random distortion means choosing a direction at random (i.e. an angle between 0 and 360 degrees) and moving the recorded location from some distance proportional to a Gaussian centered on the original location. With a standard deviation of 50 meters, we have observed that it is still quite easy to identify the house of taxi drivers with high confidence. This indicates that applying a random perturbation with a small deviation is not sufficient if the same location appears in the geolocalised dataset several times[8].

[5]It is worth noting that for protecting their privacy, we blurred their address. However, the interested reader can obviously find the actual information by applying the same algorithms we did on the original dataset.

[6]The median trace of a time window is obtained by first ordering chronologically the mobility traces contained in the time window and choosing the one located in the center position as the median.

[7]The other remaining ones can easily be eliminated by a human with the help of logic and common sense.

[8]This is the case in our study where some taxi drivers stop in front of their house every day during a period of one month.

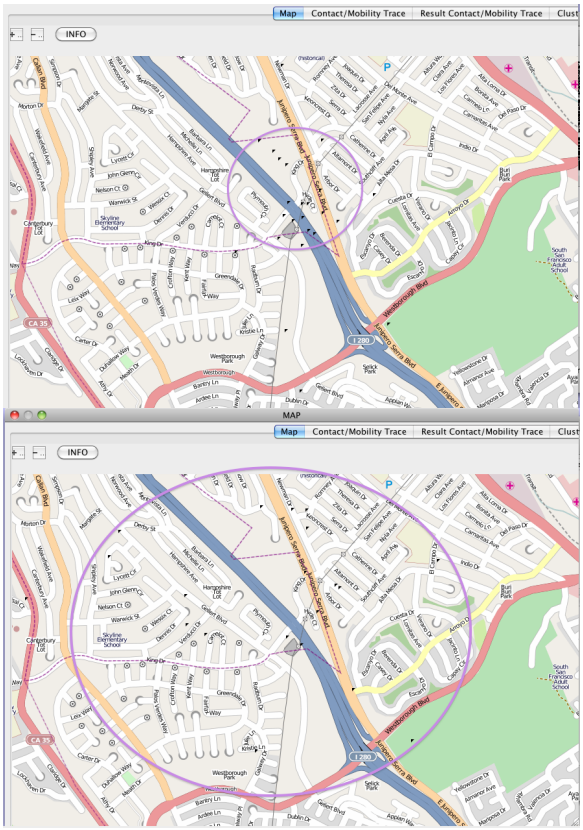

Fig. 2. Result obtained for *begin and end location finder* inference attack with a random distortion of standard deviation 200 and 1000 meters.

Starting from a deviation of 200 meters usually leads to a degradation that is high enough to makes it impossible to exactly find the house (see Figure 2). It is still possible however to detect the neighbourhood where the taxi driver lives. We also suspect that computing the median of a particular cluster corresponding to several perturbed versions of the same initial location will reveal a location that is very close to this original location. Indeed, as the median is a more robust statistic than the average, there is a high probability that it corresponds effectively to a location that has been modified only a little bit.

The random distortion has the inverse effect of the sampling on the average speed. More precisely, perturbing the reported locations of mobility traces has a high probability of increasing the distance between two consecutive traces thus increasing at the same time the average speed reported. For instance, with a perturbation of standard deviation 50 meters the average speed increase by 5% whereas it can go up to 200% when the perturbation is performed with a standard deviation of 1000 meters. An interesting direction for future work is to combined the downsampling and the random distortion is a smart manner so that they roughly cancel each other regarding their impact on the average speed.

V. Conclusion and Future Works

The spatio-temporal data of an individual is one of the most sensitive personal information and as such should be protected from falling in the hands of an unauthorized entity which could use it to cause a privacy breach. In the context where mobility traces of individuals have been collected and will be made available publicly (for instance for research or statistics purposes), it is especially important to sanitize this data before its release. However, the preservation of privacy brought by the sanitization process comes at the cost of a degradation of the quality of the data, thus also potentially decreasing its utility. GEPETO is a flexible tool for managing geolocalised data which has been especially designed to integrate into a unified approach the three aspects of sanitization, inference and utility. In our preliminary experiments on geolocalised data of taxis from San Francisco, we have studied how simple sanitization methods such as downsampling and random distortion impact the ability of an adversary to infer the house of taxi drivers and influence the utility of the resulting data. For future works, we plan to implement and evaluate more complex sanitization algorithms such as spatial cloacking and mix-zones. We also want to extend our experimentations to other types of datasets (for instance data coming from nomadic users of cell phones or containing contact traces). We will also develop more sophisticate inference attacks where an adversary tries to learn the places of interests or the social network of a particular user. Finally on the theoretical side, we want to investigate the very foundations of geo-privacy and design sound and relevant privacy and utility measures in this domain.

Acknowledgment

The authors would like to thanks Yves Deswarte and Jordi Nin Guerrero for fruitful and interesting discussions on the subject of geo-privacy.

References

[1] M.P. Armstrong, G. Rushton and D.L. Zimmerman, "Geographically masking health data to preserve confidentiality", *Statistics in Medicine* 18: 497-525, 1999.

[2] A. R. Beresford and F. Stajano, "Location privacy in pervasive computing", *IEEE Pervasive Computing* 3(1): 46-55, 2003.

[3] D. Chaum, "Untraceable electronic mail, return addresses and digital pseudonyms, *Communications of the ACM* 24(2): 84-88, 1981.

[4] *Mobility, Data Mining and Privacy; Geographic Knowledge Discovery*, F. Giannotti and D. Pedreschi (Editors), Springer-Verlag, 2008.

[5] M. Gruteser and D. Grunwald, '"Anonymous usage of location-based services through spatial and temporal cloaking", *In ACM/USENIX International Conference on Mobile Systems, Applications and Services (MobiSys)*, 2003.

[6] B. Hoh, M. Gruteser, H. Xiong and A. Alrabady, "Enhancing security and privacy in traffic-monitoring systems", *IEEE Pervasive Computing* 5(4): 38-46, 2006.

[7] A. Pfitzmann and M. Hansen, "Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity Management a consolidated proposal for terminology", Available at http://dud.inf.tu-dresden.de/Anon_Terminology.shtml, February 2008 (version 0.31).

[8] M. Piorkowski, N. Sarafijanovic-Djukic and M. Grossglauser, "CRAWDAD data set epfl/mobility (v. 2009-02-24)", Downloaded from http://crawdad.cs.dartmouth.edu/epfl/mobility, Feb. 2009.

[9] L. Sweeney, "k-anonymity: a model for protecting privacy", *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems* 10(5): 557-570, 2002.