2012 IEEE 13th International Conference on Mobile Data Management

SAWLnet: Sensitivity AWare Location cloaking on road-NETworks

Claudio Silvestri
Università Ca' Foscari Venezia
Venice, Italy

Emre Yigitoglu
TOBB University
Ankara, Turkey

Maria Luisa Damiani
University of Milan
Milan, Italy

Osman Abul
TOBB University
Ankara, Turkey

***Abstract*—Location based queries are increasingly common in mobile applications, and the associated privacy issues have become a hot research topic in the last years. Most of the current approaches, however, do not account for the location of potentially sensitive places and for constraints on the movement of users, such as speed limits or network contraints.**

In this demo we present different deployment scenarios of a privacy-preserving framework for the protection of sensitive positions in real time trajectories. We assume that the sensitivity of users' positions depends on the spatial context, while the users' movement is confined to road networks and places. Further, the users are non-anonymous, as in the case of geo-social network members who agree to share their exact position whenever it does not fall within a sensitive place, e.g. a hospital.

We will show that our proposal is suitable for different classes of devices and can be integrated in different kind of location based applications.

Device Class		Architecture
	limited memory, limited storage, slow CPU	Server side
	limited memory, large storage, slow CPU	Hybrid
	large memory, large storage, fast CPU	Client side

Fig. 1. Device classes and their peculiarities

I. Motivation

A large number of mobile applications are based on the collection of user's positions, usually up to a precision that is far greater than the one required to satisfy the user's needs. Those positions, once disclosed, could be stored for an indefinite time and represent a potential threat to the privacy of the users, specially if combined to other kind of data that allow to make unwanted inferences on the activities of the users at some given time in the past or about the belonging of the users to some social group. An example of such contextual data are the position and use of buildings in the city [1], [2]. For instance, repeated visits to healthcare facilities could reveal information on the user's health state, whereas visits to religious buildings may be a clue on her religious belief.

It is worth noticing that most existing approaches to the preservation of privacy in LBS (Location Based Services) do not directly tackle this kind of privacy threat. Instead they try to solve it indirectly, with no guarantees, by cloaking the user's position in a way that is not dependent on what we are interested in hiding. Unfortunately this could result in an excessive loss of position accuracy, or at the opposite in an inadequate protection, as in the case of an enlarged user position that is entirely contained into a sensitive location such as a hospital. The SAWL framework for privacy protection addresses this issue by accounting for both the position of sensitive sites in the territory and the statistical distribution of the user population, in order to ensure that in case a generalized user position is made public, and it contains some sensitive location, it will not be possible to assess that the user was likely inside the sensitive part of the disclosed region.

Supposing that the user movements are tied to a road network and constrained by speed limits some additional privacy threats raise, since those constraints could be used by others to restrict the user position to a much more precise location than the user would expect. For this reason, a more recent work [3] extends the contribution presented in [4] to address the same problem on a road network connecting places characterized by a typical population and a feature type, that the user may use to specify what is sensitive to her and to what extent.

In this demo we will discuss different alternatives for the deployment of the cloaking mechanism on mobile devices to achieve a usable and efficient solution. We will focus on both the different device capabilities and on the user point of view, to show how it is possible to protect the privacy of the user and to increase her privacy awareness without affecting the user experience in the use of a location based application. To emphasize the perception of the actual complexity of the system hidden by a friendly and transparent user interface, we will show in parallel what is happening in the system and what the user is seeing while using a privacy preserving location based application, such as a simple closest restaurant finder.

II. Cloaking

In [3], the user is supposed to move along a road network that connect places. These places are characterized by popu-

978-0-7695-4713-8/12 $26.00 © 2012 IEEE
DOI 10.1109/MDM.2012.72

IEEE computer society

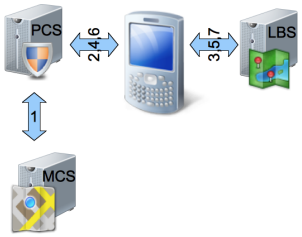

Fig. 2. Low-end devices: server side map generation and position cloaking

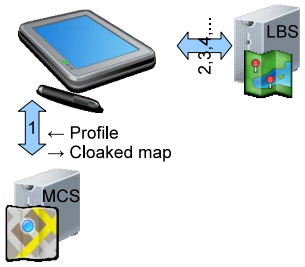

Fig. 3. Mid-level devices: server side map generation, client side position cloaking

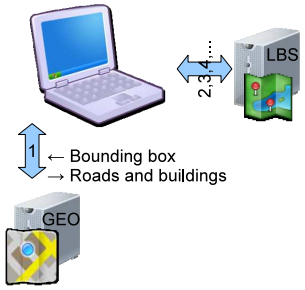

Fig. 4. high-end devices: client side map generation and position cloaking

larity values, in such a way that it is possible, given that a user is contained in one of the places from a set, to determine for each of the places the probability of finding the user in it.

Instead of disclosing their exact position, users protect their privacy by disclosing a part of the road network (and associated places) that contains their actual position. In doing so, they ensure that once a cloaked position is disclosed, it will not be possible to determine whether they were in a sensitive place with a probability higher than a threshold specified on a place type basis. A privacy profile is the set of those constraints indicated for each place type.

Further, the proposed methods deal with privacy threats based on velocity constraints, named velocity attacks. In those situations untrusted third parties could determine, by exploiting velocity constraints, that part of the cloaked region declared by the user is not reachable in the elapsed time. Thus they could restrict the actual user's position to a smaller region with respect to the one that satisfies the privacy profile.

In [3] two kinds of cloaking modes are described: offline cloaking and online cloaking. In this work we will focus on the first one, in which all the cloaked regions are pre-computed. At run time, online service requests are checked for privacy breaches (against the velocity attacks). If there are no privacy breaches, the respective cloaked region is disclosed to the LBS provider, otherwise a transformation is needed. We consider two kinds of transformations: time delay, in which the request is postponed in time domain, and postdating, in which a previously private position is disclosed instead of the current one.

The time delay mode introduces temporal error while the postdating mode introduces spatial error, both measured using temporal metrics. Unless the time delay is not greater than the acceptable time delay threshold, we prefer time delay over space error. Otherwise, we apply postdating.

III. Architectures

Mobile devices are quite heterogeneous with respect to their computational resources, chiefly CPU, RAM, and flash memory. Consequently, in the design of our system we accounted for the existence of different classes ranging from the cheapest smartphone to the more expensive tablet PC and notebooks.

For each kind of device the available resources influence the autonomy degree of the client, forcing a migration of the more computational challenging parts to a trusted third party server aware of our exact position or at least of our privacy profile.

According to this analysis we identified three different classes of smart terminals:

- **[Server side]** Low-end devices with limited memory: the client is not able either to generate a cloaked map by itself, or to store it. Thus, it is forced to rely on a position cloaking service (PCS) to cloak its position.
  Figure 2 visually shows the interaction of the client with the PCS and with the LSB. Before each access to an untrusted LBS application the client sends its current position to the PCS to obtain a position that can be safely disclosed. The trusted PCS computes the cloaking of the client position in such a way that the constraints detailed in the user's privacy profile are satisfied and sends it back to the client. Then the client can use it to access the untrusted LBS.
  The privacy profile can be cached by the PCS, as well as the partial results used for the cloaking, to simplify the answer to subsequent enquiries.
- **[Hybrid]** Mid-level devices having a large storage: the client is not able to generate a cloaked map by itself, due to its limited RAM and slow CPU. However it has enough secondary storage, for example on microSD card, to store a cloaked map for later use in autonomous cloaking. Since the client is not able to generate the map, it relies on a map cloaking server (MCS) to perform this task and retains the map as long as it can be reused.
  Figure 3 depicts the interaction of the client with the MCS and with the LSB. When the client enters a new

Fig. 5. Profile selection

zone, it sends a bounding box for the region of interest and a privacy profile to the MCS that returns a cloaked map (collection of subgraphs). This map is stored by the client and accessed before each access to an untrusted LBS application to obtain a position that can be safely disclosed.

When the client move out of the current map, or when this is likely to happen in the near future, a new cloaked map is requested. The old one is cached for a possible future reuse, and is managed according to a cache replacement policy to limit the overall memory usage.

- **[Client Side]** High-end devices having large memory, large storage, and fast CPU. The device is completely autonomous: it fetches the road network and the position of the places in the city as well as their feature types from a service like OpenStreetMap. These data are then used to compute cloaked maps and cloaked positions.

  Figure 4 illustrates the behaviour of the client in this scenario. When the client enters a new zone, it sends a bounding box for the region of interest to a feature repository (OpenStreetMap) that returns the road network and the features that are located in that zone. The client use these data, together with the privacy profile, to generate a cloaked map that will be accessed before each LBS access to cloak the current user position before its disclosure.

  Also in this case, new data will be fetched and new maps generated depending on the movement of the client.

IV. Demo story line

In this demo we will show different possible deployment scenario for SAWLnet. During the demonstration we will cover the whole life cycle of the interaction between the mobile devices, the location based service providers, the privacy enforcement provider, and the geodata provider.

A. Building the datasets

The first step for the configuration of a running system, independently of the specific device class scenario, is the creation of an annotated city network based on the roads and buildings obtained from a geodata provider, in our case OpenStreetMap[1]. After selecting the area of interest the raw data are processed to obtain the annotated city network according to our definition. This process could be done on the fly whenever data for a new region are needed, either on the mobile device or by the privacy enforcement provider depending on the class of device in use. However, it is worth noticing that it is a one time only operation whose result is identical for all users, since it uniquely depends on the selected area, thus it does make sense to cache results for future reuse. Since the result is identical for all users, it is not sensitive. Thus, from a privacy point of view, it is indifferent to perform this operation directly on the device or in some external server, provided that we trust it will not intentionally corrupt the data we are going to retrieve (the same holds for the geo-data provider).

To demonstrate this step, we will show how to obtain the dataset presented in [3], based on the OpenStreetMap's Milano Street Map.

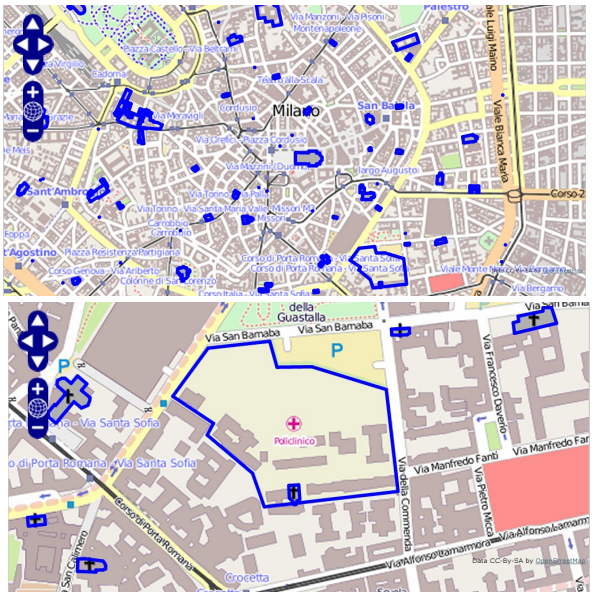

Fig. 6. Map of sensitive locations at two different levels of detail

B. Profile selection

The privacy profile is based on the user's perceptions of her own privacy, in particular on which kind of association to semantic locations she deems harmful and to which extent. In the application we will show in the demo, the user is able to either select one of the predefined privacy profiles or to define a new one, by using the graphical user interface in Figure 5.

A selection of sensitive location type determine the identification of a set of sensitive places in the city. Figure 6 shows a map of the center of the city of Milan (Italy) at two different level of detail with the sensitive locations marked in blue.

During the demo we will highlight the effects of different privacy settings on the quality of service and on the protection

[1]http://www.openstreetmap.org/

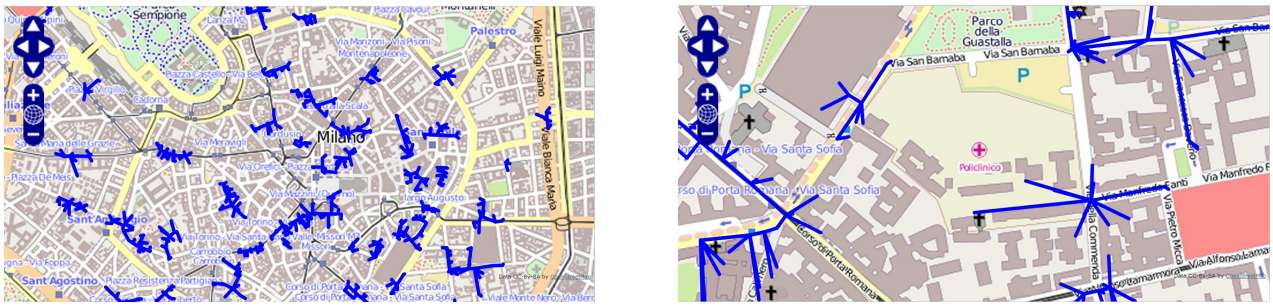

Fig. 7. Cloaked map at two different levels of detail

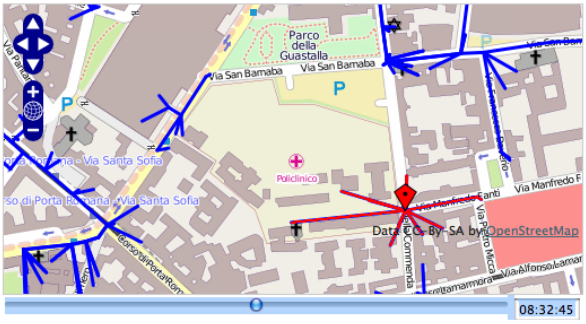

Fig. 8. History of our past positions

from velocity and location semantic based attacks, both in a qualitative way and by means of summary statistics on spatio-temporal error and position accuracy collected during the usage.

C. Walking the city

After these two preliminary steps we will start the core part of the demo, focused on the behaviour of the system as the users move in the city and use some LBS service to share their position with closeby friends and to find the nearest restaurants and shops. For the sake of the demo, we will fake the user position to simulate the movement across the city.

To give a better insight on the internals, during this part of the demo we will use a composite user interface with a part taking care of the user movement and actions, a part showing the cloaked map with highlighted current and past positions, and a part showing the emulated device interface.

The movement emulation interface allows to select the position of the user, either directly or by replaying recorded trajectories. In the second case, also past actions such as LBS requests can be replayed. As the user moves and performs actions, the application specific interface is visible in the device emulator. In the case of the restaurant finder, for instance, we will see our current exact position, the cloaked position that will be disclosed, and visual alerts of delay or postdate due to velocity attack avoidance. Finally a web interface show what is happening server-side: the position tracked by the LBS provider (Figure 8), the maps generated by the Map Cloaking Server, and the statistics about cloaking and transmitted data.

In this way we aim at exhibiting the non-intrusive integration of our framework in different kind of applications, by showing at the same time the normally hidden activities that are necessary to ensure the user's privacy in different deployment scenarios.

V. Conclusion

We developed a prototype of the privacy-preserving framework for the protection of sensitive positions in real time trajectories that we proposed in [3]. In this demo we have shown the suitability of our proposal for different classes of devices and deployment scenarios as well as the possibility of integration of the proposed approach in different kind of location based applications. From a user's point of view, the framework allows for an increase of the privacy awareness, without disrupting the user experience.

Further details on the prototype are available on the SAWLnet web site (`http://www.silv.eu/SAWLnet`).

References

[1] M. L. Damiani, C. Silvestri, and E. Bertino, "Fine-Grained Cloaking of Sensitive Positions in Location-Sharing Applications. IEEE Pervasive Computing," *IEEE Pervasive Computing*, vol. 10(4), pp. 64–72, 2011.

[2] M. L. Damiani, E. Bertino, and C. Silvestri, "The PROBE Framework for the Personalized Cloaking of Private Locations," *Transactions on Data Privacy*, vol. (3)2, pp. 123–148, 2010.

[3] E. Yigitoglu, M. Damiani, O. Abul, and C.Silvestri, "Privacy-preserving sharing of sensitive semantic locations under road-network constraints," in *Proc. MDM*, 2012.

[4] G. Ghinita, M. L. Damiani, C. Silvestri, and E. Bertino, "Preventing velocity-based linkage attacks in location-aware applications," in *Proc. 17th ACM GIS*, 2009.