Privacy issues when sharing sensitive semantic locations in urban settings

Osman Abul
Department of Computer Engineering,
TOBB University of Economics and Technology, Ankara, Turkey
osmanabul@etu.edu.tr

Host: Maria Luisa Damiani
University of Milan, Milan, Italy
mdamiani@dico.unimi.it

a MOVE-COST STSM report
Period: April 22, 2012 - May 15, 2012

May 17, 2012

Abstract

Since the trajectory data is easily produced, collected and stored the challenge now is to make it useful for various purposes. To this end many location based services are implemented and new data mining/analysis techniques are proposed for better decision making. Privacy requirements, on the other hand, hinder the development of services and analysis as the location data offers intrusive inferences about people's sensitive whereabouts. So, this makes the privacy protection an important issue when provisioning location based services and trajectory data publication. In this project, privacy issues at sensitive semantic location hiding in urban settings are studied. This project builds upon our previous collaborative work with the host. The project extended the problem formulation, attack modeling, theoretical analysis and algorithms. Most importantly, (i) optimally generating strongly cloaked regions problem is shown to be NP-Hard, (ii) two use cases of location sharing, sporadic and constant, are identified and related attack models with respective solutions are provided, (iii) a solution to off-line publication of whole trajectories are developed, and (iv) model is extended with additional background information including temporal semantics and frequency. The next step will be experimental evaluation of the new theoretical study.

Keywords: Privacy, mobility, sensitive semantic locations, location-based services

1 Introduction

This STSM project builds upon our previous collaborative work with the host. The previous collaboration resulted in a conference paper published in IEEE MDM 2012 conference [3], where the problem of sensitive semantic location sharing is studied in the context of urban settings. The urban setting suggested us to model the background information with annotated city networks and user-specific privacy profiles. Annotated city network (an undirected and weighted graph) is simply the road-network of the city plus sensitive/nonsensitive locations attached to the road network. User's privacy profile specifies the types of sensitive places and the respective quantification of desired privacy level. The solution is cloaking the annotated city network and providing a cloaked region identifier when an LBS is requested within it. A cloaking region is simply a subgraph of annotated city network which includes many cloaking regions. We used breadth-first search (BFS) to generate cloaking regions given the annotated city network and user's privacy profile. However, cloaking does not solve all kind of attacks including velocity-based attacks. To guard against such attacks we devised two methods, namely *time-delay* and *post-dating*. Time-delay delays the LBS request for some time so that no place within the cloaking region could be ruled out. Post-dating, on the other hand, answers LBS request immediately but with a user's previous safe cloaking region.

In the rest of this section, I give an overview of the previous work [3], which is necessary to explain the work, starting from Section 4, that has been done during the course of the STSM.

1.1 Background knowledge model and definitions

Let us denote with PT and P the set of place types (e.g. hospital, mall) and places (i.e. Policlinico, Carrefour) in a bi-dimensional coordinate space. The concept of *annotated city network* to model the background knowledge on the urban setting is defined next.

Definition 1 (Annotated city network) *An annotated city network is a connected and undirected weighted graph* $G{=}(V, E, pop, pt, tt)$ *where:*

i) $V = V_P \bigcup V_J$ *is the set of vertices with* $v \in V_P$ *representing a place and* $v \in V_J$ *a road junction*[1]*. The subgraph consisting only of vertices* V_J *and edges between them are*

[1]To simplify the terminology, we use the term place for both the elements in V_P and the corresponding

called city road network. *Without loss of generality, it is assumed that the city road network is connected too.*

ii) $E \subseteq V \times V$ *is the non-empty set of edges where edge* $(u, v) \in E$ *denotes a road segment connecting two road junctions or, alternatively, one road junction and one place.*

iii) Each place has a popularity and a type, expressed by the functions $pop : V_P \rightarrow (0, 1)$ *and* $pt : V_P \rightarrow PT$*, respectively*

iv) Every edge $e = (u, v) \in E$ *is assigned a weight of travel time, i.e.* $tt : E \rightarrow \mathbb{R}$*, denoting the minimum time needed to travel from* u *to* v*, and vice versa.* $\Diamond$

Note that the popularity of a place is intended to represent the prior probability that a random user is located in that place. Places having popularity 0 are places that are not reachable and thus are not relevant.

Definition 2 (region) *A* region *is a connected subgraph of the city network, denoted* $G' = (V', E')$ *with* $V' \subseteq V$ *and* $E' \subseteq E$*. The simplest region consists of a single place. In that case the graph degenerates into a singleton graph.*

Given a region r, the popularity of a place type pt in r, denoted $pop_r(pt)$, is the aggregated popularity of the places of that type located in r. Conventionally, $pop_r(.)$ denotes the popularity of the region, i.e. $\sum_{pt_i \in PT} pop_r(pt_i)$. For example the popularity of a region which only encloses roads (and no places) is 0.

Finally, the *real time trajectory* of a user over a city network is a sequence of timestamped regions, i.e. $T = \{(r_1, t_1), (r_2, t_2), \ldots, (r_n, t_n)\}$ with $t_i < t_{i+1}$. The snapshot position (r_i, t_i) means that at time t_i the user is located in the subgraph of region r_i where the subgraph can also be a singleton graph; (r_n, t_n) is the current position.

1.2 Privacy requirements on single Cloaking Regions

The privacy profile specifies for each place type $pt_i \in PT$ a user-defined threshold value τ_i indicating the maximum allowed probability of association between a user and a place of such type. Formally, the pair (pt_i, τ_i) prescribes that in any CR the posterior probability

locations

that a user is in a sensitive place of type pt_i must not exceed the user-defined threshold. The notion of *strongly cloaked region* is defined next.

Definition 3 (Strongly cloaked region) *A strongly cloaked region r, for a given privacy profile, is a region satisfying the following conditions:*

- *r contains at least one sensitive place*
- *The popularity of r satisfies the following inequality :*

$$\sum_{pt_i \in PT_S} \frac{pop_r(pt_i)}{\tau_i} \leq pop_r(.) \tag{1}$$

$\Diamond$

1.2.1 Cloaking Map Generation

The pseudo-code of cloaking map generation is shown in Algorithm 1, respectively. Starting from the sensitive seed node, the algorithm does a breadth-first search (BFS) to extend the subgraph for the respective cloaked region. BFS is preferred because it tends to output compact (small diameter) subgraphs. After the BFS traversal is completed, all the original edges between the vertices are included in the resulting BFS tree. This is particularly important to preserve the shortest paths among the vertices of the subgraph. The output of the algorithm is called the cloaking map, consisting of a number of cloaked regions per profile.

1.2.2 Velocity Attack and Transformation

When the user requests to use LBS with his last position, then velocity attack should be voided. Algorithm 2 gives the pseudo-code for the transformation needed to guard against the velocity attack. First the algorithm checks whether it is possible to convey the request under the velocity attack; if so, the current position is said to be safe with respect to the previous position and is conveyed (with or without cloaking depending on the location of the user w.r.t. the cloaking map). If the current position opens up a privacy breach, computed in line 5 according to Equation 5, then two alternatives (time delay and postdating) are evaluated and time delay is preferred in case the best time delay is less than a predefined maximum delay parameter. In case it is impossible, postdating remains the only possibility.

Algorithm 1 Cloaking map generation

Require: Annotated city network $G = (V, E, pop, pt, tt)$, privacy profile $PP = \{(pt_i, \tau_i)\}_{i \in [1,n]}$

Ensure: Cloaked region map

```
1: map ← ∅
2: for all u ∈ V s.t. u.pt ∈ PT_S do
3:    cr ← ∅
4:    totalPop ← u.pop
5:    while (true) do
6:       v ← next move from BFS(u)
7:       if v.pt ∈ PT_NS then
8:          cr.addEdge(edge(parent(v), v))
9:          totalPop ← totalPop + v.pop
10:         if u.pop / pop_cr(·) ≤ τ_i where u.pt = pt_i then
11:            break
12:   map ← map ⋃ {cr}
```

In case the postdating introduces too much spatial error, then it might be preferable to drop the service request rather than reporting an obsolete location.

2 Purpose of the STSM

The purpose of the STSM was to extend the previous work [3] in a few directions, as outlined next.

- Analyzing deeper theoretical properties of the problem and proofs of different aspects for our proposal.
- Extending the work in a few directions, e.g. scalability, minor variations of problem setting and developing more effective/efficient algorithms.
- Shifting the privacy protection of trajectory data from location based services context to data publishing context, e.g. anonymization of spatio-temporal sequences.

With these extensions, the project aimed to come up with an extended version of the paper which can be published in a pioneering data engineering journal.

Algorithm 2 Transformation

Require: Annotated city network $G = (V, E, pop, pt, tt)$, cloaking map map, request timestamp t_q, location loc of user U

Ensure: Cloaked region/point and issuance time

Let A to be last issued cr/point with issuance time t_A
$CRsU \leftarrow \{cr \in map : loc \in cr\}$
if $CRsU = \emptyset$ **then**
  $CRsU \leftarrow loc$ ▷ *a single point cr*
$\overline{CRsU} \leftarrow \{cr \in CRsU : cr\ is\ safe\ w.r.t.\ A\}$
if $\overline{CRsU} \neq \emptyset$ **then**
  return a random $cr \in \overline{CRsU}$ and t_q
$mindelay \leftarrow min_{cr \in CRsU}\{delay\ needed\ for\ cr\}$
if $mindelay \leq MAX\_DELAY$ **then**
  ▷ *time delay*
  $cr_{min} \leftarrow argmin_{cr \in CRsU}\{delay\ needed\ for\ cr\}$
  return cr_{min} and $t_q + mindelay$
else
  ▷ *postdate*
  $cr_f \leftarrow$ first safe cr (w.r.t. A) along regressing $path(loc, A)$
  return cr_f and t_q

3 Description of the work carried out during the STSM

3.1 Properties of the Problem

Property 1 *A strongly cloaked region is still a strongly cloaked region if positive popularities are assigned to roads.*

Proof sketch. Following the commonsense that all roads are nonsensitive, positive popularity for roads only increases the right side of inequality 1.

This property also justifies whey the roads in our model are assigned zero popularity even though they are usually populated.

Property 2 *Between any pair of places, there exists a path passing through no intermediate places.*

Proof sketch. The underlying city road network is connected and every place is directly linked to the city road network.

Theorem 1 *Minimum diameter strongly cloaking region generation is NP-Hard.*

Proof. Our proof is base on a reduction from Clustering problem shown to be NP-Hard [2]. Instance of the clustering problem includes a finite set of objects $X = \{x_1, x_2, \ldots, x_n\}$, a positive integer distance function $d(x, y)$ defined for any $x, y \in X$, and a positive integer B. It asks whether there is a partition of X into three disjoint subsets X_1, X_2, X_3 with which, for each set X_i, for all pairs $x, y \in X_i$, it holds that $d(x, y) \leq B$. Given an instance of the clustering problem, the reduction is as follows.
$V = X \bigcup \{y_1, y_2, y_3\}$, E is a clique over V, and the weight function (the function tt) is defined as: for $(u, v) \in E$ (i) $tt(u, v)$ if both $u, v \in X$, (ii) $tt(u, v) = 3B$ if $u = y_i$ and $v = y_j$), s.t. $i \neq j$, (iii) $tt(u, v) = 1$ otherwise (i.e. one is x_i and the other is y_i). For $v \in V$, pt is defined as: (i) $pt(v) = Sensitive$ if $v = y_i$, or (ii) $pt(v) = Nonsensitive$ if $v = x_i$, i.e. there are three sensitive places and n nonsensitive places. For $v \in V$, pop is defined as follows: (i) $pop(v) = 0.5$ if $v = y_i$, and (ii) $pop(v) = 0.5\frac{3}{n}$ if $v = x_i$. Finally, we let $D = B$ and privacy profile be $\tau_{y_i} = 0.5$ for all $i = 1, 2, 3$. And also assume without loss of generality that n is a multiple of 3.

Forward direction. Suppose there exist a solution X_1, X_2, X_3 to Clustering problem, then there exists three cloaking regions cr_1, cr_2, cr_3 solution to Cloaking problem. cr_1 contains all the nodes from X_1 and y_1, cr_2 contains all the nodes from X_2 and y_2, and cr_3 contains all the nodes from X_3 and y_3. Note that diameters of X_i and cr_i are the same as y_i has no effect on the diameter. Moreover, due to the construction inequality 1 is satisfied for any cr_i as popularity of y_i and the sum of popularities in X_i is a parity and hence obeys the privacy profile.

Backward direction. Suppose there exists to a solution to Cloaking problem. The solution cannot place the three sensitive places into the same cloaking region as their pairwise distance is greater than D. So, there should be exactly three cloaking regions each containing one sensitive place. For each cloaking region there should be $n/3$ nonsensitive place inside as otherwise the inequality 1 cannot be satisfied. Since each cloaking region has diameter not greater than D, there exist a Clustering solution with cost at most $B = D$.

Efficiency. The reduction is a linear algorithm, i.e. runs in $O(V + E)$ time.

[2] http://www.csc.liv.ac.uk/ ped/teachadmin/COMP202/annotated_np.html

◇

3.2 Extending the Work

3.2.1 Location sharing/LBS request modes and Attack models

Location sharing and LBS requests are indeed distinct entities and for our purposes four possible modes as presented in the table below are considered.

Mode	Location sharing	LBS request
Mode 1	Constant	Constant
Mode 2	Constant	Sporadic
Mode 3	Sporadic	Constant
Mode 4	Sporadic	Sporadic

In the Mode 1, the user shares his location all the time and request LBS service all the time. In the Mode 2, although the user shares his location all the time he requests LBS service sporadically. Mode 3 and Mode 4 are defined similarly. From the attackers point of view, Mode 1, Mode 2 and Mode 3 are the same as they all give the actual trajectory of the user. Mode 4 is different as it gives only a few samples along the trajectory. Due to this equivalence we name the first three modes as *constant location sharing* and the last mode as *sporadic location sharing*. In fact, there are real world applications for both modes of the location sharing.

Constant Location Sharing. The privacy requirements given in the previous subsection tell us that users can share their locations if they are not entering any of the cloaking regions, hence not visiting any sensitive place. Indeed some LBS applications require users to constantly share their locations in case it does not cause any privacy breach for them. So, this suggests that in this mode the adversary knows the real time trajectories of the users. However, care must be taken when a user trajectory involves passing through cloaked regions as well. In this case, when the user enters into a cloaking region he stops location sharing and whenever he needs an LBS service just sends the cloaking region ID. Upon exit from the cloaking region he resumes constant location sharing. We name this kind of use mode as *constant location sharing*.

Property 3 *With constant location sharing mode of use, the attacker not only knows the entry and exit times to cloaking regions but entry and exit points as well.*

As a result of the above property all the boundaries (entry and exit vertices) of the cloaking regions must be nonsensitive. Moreover, there should be a path between the entry and exit vertices involving no intermediary sensitive place. Otherwise, the attacker will simply reason that the user passed through the sensitive place due to lack of alternative.

Property 4 *If every entry and exit vertices of a cloaking region cr are road junctions and the subgraph cr is still connected after removing sensitive places, then lack of alternative attack is voided.*

Proof sketch. All the entry and exit vertices are nonsensitive and there exist a path between any pair not involving any sensitive place.

Consider that a user enters only one cloaking region cr during the course of the trajectory, with entry time t_e, exit time t_x, entry location $v_e \in cr \in V$ and exit location $v_x \in cr \in V$.

Let $sp_{(v_e,v_x)}(v)$ be the shortest path between v_e and v_x passing through the place $v \in cr \in V_P$. And let $EX_{(t_e,t_x)}$ be the places with the shortest path not greater than $t_x - t_e$, i.e. $EX_{(t_e,t_x)} = \{v : v \in cr \ s.t. \ sp_{(v_e,v_x)}(v) \leq t_x - t_e\}$. Clearly, the user can only visit the places in $EX_{(t_e,t_x)}$ and cannot visit the places in $cr \setminus EX_{(t_e,t_x)}$. A privacy breach occurs when inequality 1 cannot be satisfied for the region containing points only from $EX_{(t_e,t_x)}$. This is simply because the user's privacy profile is violated. To respect the user's privacy profile we need to delay t_x till the smallest t_r, called effective exit time, such that $EX_{(t_e,t_r)}$ meets the inequality 1. Note that such a $t_r \geq 0$ always exists as all the in cr places are reachable from both v_e and v_x. The time $t_r - t_x$ is the minimum time to be compensated, i.e. on exit the location sharing should commence only after this time can be compensated along the way.

Note that in the above setting no LBS service request is considered. In the same setting, suppose that the user request a service request at time $t_s \in [t_e, t_x]$, i.e. inside cr. Let $sp_{(v_e}(v)$ be the shortest path between v_e and v. And let $ES_{(t_e,t_s)}$ be the places with the shortest path not greater than $t_s - t_e$, i.e. $ES_{(t_e,t_s)} = \{v : v \in cr \ s.t. \ sp_{(v_e,v)}(v) \leq t_s - t_e\}$. Similarly, the user can reach to places in $ES_{(t_e,t_s)}$ and cannot reach the places in $cr \setminus ES_{(t_e,t_s)}$. A privacy breach occurs when inequality 1 cannot be satisfied, i.e. privacy profile violation, for the region containing points only from $ES_{(t_e,t_s)}$. Again, to respect the user's privacy profile we need to delay t_s till the smallest t such that $ES_{(t_e,t_s)}$ meets the inequality 1. The service request should be delayed by $t - t_s$. If the delay is positive, then a *time delay* can be applied.

Property 5 *As a special case if the user cannot visit any sensitive place in the cloaking region, then:*

(1) the compensation time is zero, and

(2) if an LBS is requested the time delay is zero.

Proof sketch. Left-hand side of inequality 1 is zero and hence it is satisfied.

Property 6 *The effective exit time is an upper bound for the actual service request time. As a result, the service request is delayed at most* $t_r - t_s$.

Proof sketch. Simply because effective exit time ensures every place can be visited and hence inequality 1 satisfied.

The above property ensures that service requests inside the cloaking region does not adversely affect the performance of location sharing quality.

For the consecutive cloaking regions the two cases are possible, (i) effective exit time from the first is not greater than the entry time to the second, (ii) effective exit time from the first is greater than the entry time to the second. The former case simply states that the two cloaking regions are decoupled w.r.t. entry and exit times and hence can be handled independently as already explained. But for the latter case, care must be taken as there is a coupling. However, the solution is simple: just add the positive difference, effective exit time from the first minus the entry time to the second, to the effective exit time from the second. Note that the difference is the upper bound as some of the difference could be already compensated along the way from first to second cloaking regions. The difference can be considered like a carry to be accumulated on the independently computed effective exit time.

Suppose a user visits two or more cloaking regions before the compensation time reaches zero, then a *post-dating* could be applied in case service request is issued any time after leaving the first cloaking region. In other words, after that time all service requests can report the identifier for the first cloaking region in the sequence. Clearly, when both time delay and post-dating are applicable, then one should be preferred. As a generic solution we prefer time delay if the delay is less than a user-specified threshold, otherwise we apply post-dating.

Sporadic Location Sharing. Given a set of cloaking regions, the user may wish to share his location either (i) precisely when out of any cloaking regions, or (ii) coarsely when inside a cloaking region. With the former, even though location sharing is safe care must still be taken. For instance, the shared position can be just before entering a cloaking region, hence in this case the attacker may know the entry point to the cloaking region and approximate time of entry. For this reason, similar to the constant location sharing, all the entry/exit points of cloaking locations should be nonsensitive places, e.g. road-junctions. For the latter, only the cloaking region identifier is shared but a time delay or post-dating may be needed as discussed next.

The effectiveness of the cloaking method can be compromised by the velocity-based linkage attack [1], i.e. an adversary can leverage the information on the maximum velocity to delimit the user's position within the CRs reported in the shared trajectory. We recall that the edges of the city network are weighted with travel time, expressing the minimum time (i.e. maximum velocity) to traverse an edge and that such information is publicly known. To prevent this privacy breach, we redefine the *safety* condition that must hold between two consecutive CRs or one CR and one precise location for a shared trajectory not to be susceptible to velocity-based linkage attacks [1].

Accordingly, we define the node-pairwise distance $d_{pp}(G_1, G_2)$ between the two regions G_1=(V_1, E_1) and G_2=(V_2, E_2) as the longest shortest path between any node in G_1 and any node in G_2, i.e. $d_{pp}(G_1, G_2) = \max_{v \in_l V_1} \max_{w \in_l V_2}$ *ShortestPath(v,w)*. Notice that the distance along the graph is measured in time units. If one of the regions is a precise location then the respective subgraph is a singleton. The safety requirement is as follows: G_1 and G_2 are safe to disclose if the node-pairwise distance between them is lower than the time t spent by the user to reach G_2 from G_1 (or vice versa), i.e.:

$$d_{pp}(G_1, G_2) < t \tag{2}$$

If the service request is made t_s units later on entering region G_1, then the time delay is $t - t_s$. We call the safety requirement given in inequality as *stringent safety*. The safety requirement ensures that none of the involved CRs can be shrank with velocity-based linkage attacks, as inequality 1 satisfied. However, the time delay $t - t_s$ can be big, so we look for the ways of relaxing the stringent safety requirement to improve the service quality as given next.

If one of the regions is a CR and the other is a precise location (i.e. a singleton region) then

the approach taken for constant location sharing can be taken. That is find the shortest delay time such that inequality 1 is satisfied for the subset of places in the CR. On the other hand, if both of the regions are CRs ($cr1$ and $cr2$ in order), then to answer an LBS request in $cr2$ a similar strategy can be applied. However, in this case there are many shortest path to a place in $cr2$ from any place in $cr1$. Clearly, if maximum of shortest paths for points in $cr2$ are used, the problem reduces to the former case. We call the safety requirement here as *relaxed safety*.

The time delay is still big even after the relaxed safety requirement, then a post-dating could be applied.

3.2.2 Extensively Annotated city network

Although the annotated city network (Definition 1) is an acceptable model for background knowledge, attacker may have additional information regarding the annotated city network model and attack accordingly. Consider for instance that somebody spends too much time in a cloaking region and only one sensitive place can worth spending too much but other places are almost transient, i.e. inhomogeneity of staying durations. Clearly in this case, the attacker may infer that the user has visited the sensitive place. Similarly, we can also talk inhomogeneity of usage times, i.e. some places are more frequented in evenings than in mornings.

As an another kind of attack, suppose a user enters a few times to a particular cloaking area during the course (i.e. a whole day) of the trajectory, and also suppose that none of the nonsensitive places worth to be visited multiple times. Again, the attacker may clearly reason that the user is indeed visiting the sensitive place. To guard against such kind of attacks we extend the Definition 1 with relevant information as provided next.

Definition 4 (Extensively annotated city network) *An extensively annotated city network is essentially an annotated city network with two additional background knowledge, temporal semantics $tempsem$ and typical frequency $tfreq$ of places:*

i) temporal semantics-staying duration, $tempsem : V_P \rightarrow Pr_{V_P}(X)$ is probability density function for staying duration at V_P where $X \in \mathcal{R}^+$.

ii) temporal semantics-usage time, $tempsem : V_P \rightarrow Pr_{V_P}(X)$ is probability density function for usage time of V_P where $X \in \mathcal{R}^+$.

iii) typical frequency, $tfreq : V_P \rightarrow N^+$ is the number of typical visits to places during the course of the trajectory. $\Diamond$

The work by Lee et al. [2] introduced the staying duration/usage time as means to define temporal semantics for a particular place/place type. In their framework, each place type is associated with probability density functions which represent the typical staying duration and usage times. Given two locations, their similarity (w.r.t. the staying duration or usage time) is obtained by measuring the (a modified version of) Kullback-Leibler divergence between the two probability distributions. Recognizing that that usage time is all about dynamically assigning popularity to locations, it can be handled by appropriately modifying the popularity information associated with the places depending on the time of the day, day of week etc. Hence, when there is a change, the cloaking map can be re-created with the new popularity assignments.

Although the usage time can be handled in a very straightforward way, staying duration can not be handled using the same approach. This is simply because a very frequented (hence more popular) place need not necessarily have longer staying duration. Hence we consider that popularity and staying durations are not convertible. Suppose a cloaking region is homogenous w.r.t. the staying duration of places inside it, then the attacker can not exploit this background information to localize the user within the region by simply looking at the staying duration inside it. However, it may be impossible to provide pure homogeneity as all the staying durations can be distinct. So, it suggests that staying duration is a soft-constraint with the aim of obtaining homogenous cloaking regions as much as possible. One solution is to handle this as a preference during cloaking map generation, i.e. order the neighboring places w.r.t. staying duration similarity to the seed place and consider expanding the cloaking region in this order. This is our canonical solution to exploit staying duration.

Given the typical frequencies and cloaking map, one can easily tell for any cloaking region the number of visits that are secure. For instance consider that a *cr* contains four places $< S_1, S_2, NS_1, NS_2 >$, two sensitive and two nonsensitive, with the typical frequency vector $< 2, 8, 6, 5 >$ and popularity vector $< 0.5, 0.5, 0.5, 0.5 >$ and assume that privacy profile is $< \tau_{S_1} = 0.5, \tau_{S_2} = 0.5 >$. Clearly, there is no problem until the number of visit is two however when the *cr* is visited third time we can safely change the popularity vector as $< 0.0, 0.5, 0.5, 0.5 >$ as S_1 cannot be visited in the third time. The only sensitive place now is S_2 and the last vector meets the privacy requirement. Since, the privacy requirement

is satisfied, there is no problem in the third visit. Similarly, till the sixth visit the last popularity vector is valid but after which it becomes $< 0.0, 0.5, 0.5, 0.0 >$. At this time the privacy requirement is still satisfied, but one more visit makes the vector $< 0.0, 0.5, 0.0, 0.0 >$ and the privacy requirement cannot be satisfied at the 7th visit to cr. The problem now is what happens if the user visits the cr one more time, then clearly it is obvious that the privacy is breached if the location reported is cr. Our solution is based on dynamically searching for a bigger cloaking region by taking the cr as the seed. Fortunately, our cloaking algorithms work in this situation if they are modified as follows. When a cr looses its cloaking region property start a search from the boundary of the cr and extend it till the privacy requirement is satisfied again. Note that when extending, both regions with singletons or other cloaking regions are considered. The only care taken is popularity vector of places in any cloaking region is updated after every visit to it. In summary, whenever needed we change the cloaking map dynamically and incrementally.

3.3 Trajectory data publishing

Data mining and some other data-centric applications require complete user trajectories, and have quite different notion in comparison to online service-centric LBSs. That is, there is no LBS request and location sharing during the course of the trajectory generation, but the recorded trajectory is shared when complete in an off-line fashion. Clearly, the sharing should be done in a privacy-preserving way as we assume that the owner of the trajectory is not anonymous and the user can visit sensitive places along the way.

The techniques for constant location sharing (with no LBS request), explained in Section 3.2.1, are directly applicable for trajectory data publishing. That is, if the trajectory passes through a CR cr, then between the entry and effective exit times of cr the respective region identifier is published, otherwise the precise location is published. So, the published trajectory consists of episodes of cloaking region identifiers and precise locations. Note that, the published CR identifiers give no false information but give true information in a coarse way, hence the data mining applications can safely use this kind of data, i.e. no resulting pattern would be fake.

4 Description of the main results obtained

From the work that has been explained in the previous section, our main results are as follows.

- The rationale behind accepting that the roads have no popularity is justified.
- Optimally solving the problem of strongly cloaked region generation is NP-Hard. Hence, good heuristics and search strategies are needed to attempt to solve the problem.
- The previous work is extended in a few ways. (i) separating the location sharing and location based services, (ii) considering two modes of use cases, namely constant and sporadic location sharing, (iii) privacy requirements and generic solutions for each of the two cases, (iv) extensively annotated city network as a new kind of background knowledge modeling.
- Besides BFS, other search strategies can also be used for effective finding of cloaking regions. To this end, uniform-cost search is considered to be a good choice as it always takes into account the distance from the seed vertex.
- Proposing a solution to the trajectory data publication problem.

5 Future collaboration with host institution

During the STSM, me and host (Dr. Damiani) have extensively collaborated and the results presented in the previous sections are due to our intensive discussions. During our discussions, we have identified some other problems as a future collaboration. The problems include the privacy issues in own location determination and sensitive co-location identification avoidance.

The collaboration on the current work will be continued as well. For instance, we will do an experimental evaluation of the all extensions here.

6 Foreseen publications

After completing the experimental evaluation, we hope to submit the work to a leading knowledge engineering journal.

References

[1] 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.

[2] Byoungyoung Lee, Jinoh Oh, Hwanjo Yu, and Jong Kim. Protecting location privacy using location semantics. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 1289–1297, 2011.

[3] E. Yigitoglu, M.L. Daminai, O. Abul, and C. Silvestri. Privacy-preserving sharing of sensitive semantic locations under road-network constraints. In *Proc. IEEE Mobile Data Management (MDM 2012)*, 2012.