PrivLBS: Preserving Privacy in Location-Based Services

Eduardo R. Duarte Neto, André L. C. Mendonça, Javam C. Machado

Universidade Federal do Ceará, Fortaleza, Brazil
{eduardo.rodrigues, andre.luis, javam.machado}@lsbd.ufc.br

Abstract. Location-based services have been increasingly integrated into people's daily activities. However, some of these services may not be trustworthy and lead to serious privacy breaches. While spatial transformation techniques such as location perturbation or generalization have been studied extensively, many of them only consider the location at single timestamps without considering temporal correlations among the locations of a moving user, leaving the user's location with no guarantees of privacy protection against attacks that would exploit this vulnerability. This work proposes a new technique for preserving data privacy, named PrivLBS, which ensures that the individual's location will not be easily re-identified by malicious services. Extensive simulation experiments have been carried out to evaluate the efficiency of PrivLBS. Experimental results show that PrivLBS reaches higher protection compared to other related approaches over different kinds of attacks.

Categories and Subject Descriptors: H.2.8 [**Database Applications**]: Spatial databases and GIS; K.4.1 [**Computers and Society**]: Privacy

Keywords: location-based service, location privacy, obfuscation, dummy location

1. INTRODUCTION

Location-Based Services (LBS) provide additional features to mobile devices based on the user's geographic locations. These services have been integrated into people's daily activities, enabling them to use their current location for a variety of purposes, such as navigation, searching points of interest, tracking, recommending systems, and more. In traditional LBS, users send to the service provider (LBS provider) their identity and current geographic location, defined by its latitude and longitude, in addition to requests, such as, for instance: the nearest supermarket or restaurant from the current location [Niu et al. 2014]. In this way, users get the answer regarding the performed query.

On the other hand, the use of LBS can lead to serious privacy breaches due to malicious or unreliable service providers. Untrusted LBS providers can expose location data of their users or even sell location information to third parties. The data obtained may be used to discover patterns of user trajectories, which can reveal sensitive information about the users [Brito et al. 2015]. For example, if a user, enjoying an LBS, frequently displays his or her location near a hospital, the location information could be used to infer that the user is likely to have a health problem.

Thus, privacy models are necessary to anonymize the user's request, hiding the user's location from the service provider. Several techniques have been proposed to guarantee the user's privacy in the use of LBS [Niu et al. 2016; Tsoukaneri et al. 2016; Ullah and Shah 2016; Sun et al. 2017]. Some of these techniques apply the obfuscation model [Hubaux et al. 2011; Ghinita 2013], reducing the accuracy of the actual location sent in the request. However, this approach results in data utility loss, which harms service quality. Therefore, managing this trade-off between the privacy of individuals and the data utility becomes another major challenge.

This research was partially supported by CNPq (under grant number 32614/2017-0) and CAPES (under grant number 1792482) - Brazil and LSBD/UFC.
Copyright©2019 Permission to copy without fee all or part of the material printed in JIDM is granted provided that the copies are not made or distributed for commercial advantage, and that notice is given that copying is by permission of the Sociedade Brasileira de Computação.

An alternative to the obfuscation approach is the Dummy Location technique [Kido et al. 2005], which guarantees user privacy by adding false locations (dummies) to be sent along with the real location in the request made to the service provider. In this approach, $k-1$ fake locations are selected and added to the user's query in order to hide the real location of the individual who performed the query. For instance, the moment a user wants to get the closest mall to his or her current location, by specifying the k parameter, other $k-1$ dummies will be automatically chosen and sent to the service provider. In response, the service provider will return the nearest shopping centers for each of the k locations in the request, which will include the nearest shopping center to the user's real location.

Although the technique of Dummy Location seeks to ensure proper levels of privacy without affecting the quality of the service, we identify in the existing works in the literature [Kido et al. 2005; Vu et al. 2012; Niu et al. 2014; Sun et al. 2017] a potential fragility against attacks that exploit adversary knowledge about the information present in the request, the so-called attacks of knowledge. This fragility is significantly increased when the user is conducting continuous requests to the service provider, that is, several consecutive queries over time. Thus, the main contributions of this article are:

- A knowledge attack that seeks to reveal the real location of the user present in the request. Through our attack, we demonstrated the vulnerability of the Dummy Location techniques.
- PrivLBS: a new technique based on the obfuscation model able to ensure that the locations of individuals using location-based services are not easily reidentified. Assuming that the LBS provider is not reliable, PrivLBS seeks to anonymize the request, using the provider's knowledge to ensure the privacy of the users' location.

The remainder of this article is organized as follows. Basic concepts regarding to LBS are presented in Section 2. Section 3 presents the related work to the topic of privacy preservation in location-based services. Section 4 presents our attack proposal to demonstrate the vulnerability of the related works. In Section 5, we introduce the PrivLBS method as the solution to the problem. In Section 6, we describe the performed experiments and present the results obtained. Finally, Section 7 presents our conclusions and the future directions of research.

2. BASIC CONCEPTS

In a typical LBS, Figure 1, users are participants who consume the service sending requests to the service provider using mobile devices, i.e., smartphones, notebooks, wearables, among others. Those devices have positioning systems, for instance, GPS, that allows identifying the location of the objects involved, i.e., POIs, users, or any other entity.

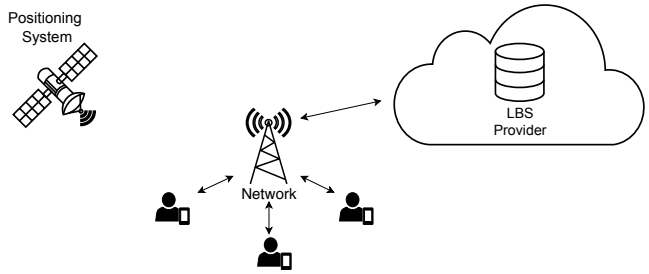

Fig. 1. Location-based services architecture.

The LBS provider is responsible for receiving the user's requests and responding according to the locations present. Since the LBS provider has access to all requests received, it may store all the

information about the locations present in the request. This knowledge can be used to retrieve sensitive information about users. In the model adopted [Vu et al. 2012], the LBS provider has the responsibility to store and make available to the users the side information [Ma et al. 2013] acquired about the locations covered, which will be used during the anonymization process.

We can define a location in this work as a point of interest (POI), containing its coordinates, popularity, and class. The coordinates, represented by the georeferencing pair (lat, lon), register the POI's real position. Each location has a class, which represents the location's category, like a hospital. The location's popularity refers to the probability of the location l to be submitted to the LBS over all locations in L, the set of locations covered by the LBS. The location probability is obtained through the Equation 1. The side information collected in our model is restricted to the location's probability of each location in L.

$$p_i = \frac{number\ of\ requisitions\ over\ l_i}{total\ number\ of\ requisitions} \tag{1}$$

A user request or query is composed of a location information and a request content. The location information is all information regarded to the user location present in the query. The request content is the query itself, for instance, "the closest hospital". In general, a request content is associated with a location class, as in the previous example, where the request's content is related to the class hospital. Figure 2 illustrates a request example.

A malicious agent, also called an adversary or attacker, seeks to exploit users' private data, such as their location. In this work, we consider the context where the service provider itself is the attacker.

Fig. 2. A typical request sent to a LBS provider.

3. RELATED WORK

Several solutions have been proposed to ensure the privacy of users in LBS and thus prevent their sensitive information from being discovered. In general, the solutions are divided into approaches based on location anonymization [Gedik and Liu 2008; Ying and Makrakis 2014; Duckham and Kulik 2005; Bamba et al. 2008] , encryption [Lu et al. 2014], obfuscation with differential privacy [Andrés et al. 2013; Wang et al. 2017], or dummy location selection [Kido et al. 2005; Niu et al. 2014; 2015; Sun et al. 2017].

Gedik and Liu [2008] propose a custom k-anonymity model using the cloak strategy. In this paper, the authors used a reliable anonymization server that considers the trade-off between location privacy and Quality of Service (QoS) to anonymize the users' location. In the solution, the algorithm forms a cloak region containing other geographically distributed $k-1$ users, before submitting the query to the location-based service. Also using the cloak strategy, Ying and Makrakis [2014] ensure users' privacy by building a camouflaging region containing at least k users and l street segments.

Lu et al. [2014] present PLAM, a privacy-preserving framework for local-area mobile social networks. The framework not only meets the k-anonymity privacy model but also ensures the l-diversity

[Machanavajjhala et al. 2006] model, considering cases where an opponent can infer sensitive information about the users without even identifying them. However, the trusted anonymization server is replaced by an encryption technique called pseudo-ID, which does not maintain the data utility for analysis purposes.

Andrés et al. [2013] and Wang et al. [2017] define an area of indistinguishability of radius r, where noise is added to the location of the user within this area. The amount of noise required to ensure user privacy is calculated through a differential privacy mechanism, reducing the data utility.

To ensure the utility of the data without allowing the LBS provider to identify the user's real location, Kido et al. [2005] propose the use of dummies to hide the real location present in a query.

3.1 Dummy Location

As already presented in Section 1, the Dummy Location technique selects $k-1$ dummies, as false locations, to be sent along with the actual location in the request made to the service provider. The main virtue of this technique is the absence of data utility loss. Also, it does not require the use of a trusted server to anonymize the requests. Each user, through a mobile device, is responsible for anonymizing the requests. As a negative point, the number of locations present in the request may generate an overhead due to the cost of selecting the dummy locations.

Niu et al. [2014] propose the Dummy Location Selection (DLS), an entropy-based dummy location selection algorithm, which measures the degree of uncertainty over a set of selected locations. In this work, the authors presented an LBS model in which the service provider is responsible for collecting and making available queries' statistical data to the users. In special, the probability of a location being presented in a query. Thus, the DLS aims to reach the k-anonymity degree of privacy by submitting a query containing the user's current location and other $k-1$ dummy locations, which have a probability of being sent to the LBS similar to that of the user's real location. First, it selects the $2k$ locations with probabilities closest to the user's real location. From those are generated m possible requests, each one with $k-1$ dummy locations besides the user's real location. The parameter k and m are defined according to the privacy requirements. A greater k and m ensures higher protection, although it increases the process overhead. From the m possible requests, it is selected the one with the highest entropy, calculated in function of the locations' probability of being present in a request. Equation 2 shows how the entropy H of a request is calculated based on the locations' probabilities q_i. The highest value H can assume is $H_{max} = \log_2 k$, which occurs when all selected locations have the same probability as the real user's location.

$$H = -\sum_{i=1}^{k} q_i . \log_2 q_i \tag{2}$$

Finally, Sun et al. [2017] propose the Dummy Location Privacy (DLP). Like the DLS, it uses the dummy location technique and the probability of the locations as a selection criterion. The DLP uses a greedy approach selection, aiming to find out the optimal set of locations to compose the query in terms of entropy, reaching as a consequence, a greater degree of uncertainty over the set of selected locations. In the DLP selection process, it guarantees that for each dummy location selected, the entropy achieved is the highest possible. In other words, if the DLP has already selected i locations ($i < k$), in the selection of the $(i+1)^{th}$ dummy location, it ensures that the entropy H_{i+1} is the highest for all the remaining locations. H_{i+1} is measured by Equation 3, where p_j is the probability of the location j. Besides, the authors propose an attack algorithm, ADLS, developed specifically to reveal the user's real location when the anonymization consider as a selection criterion of the $k-1$ dummy locations, the probability of each location to be present in a query sent to the LBS. Their results demonstrated the vulnerability of anonymized DLS requests under the ADLS attack.

$$H_{i+1} = -\sum_{j=1}^{i+1} \frac{p_j}{\sum_{l=1}^{i+1} p_l} \log_2 \frac{p_j}{\sum_{l=1}^{i+1} p_l} \quad (3)$$

Although the DLS and DLP algorithms seek to guarantee user's privacy without loss of service quality, we identify two potential risks of disclosure. First, since the dummy location approaches only act in the location information, the request content is still vulnerable to inferences about its content, which would be used for disclosure of sensitive data about the user. Another potential risk can be identified in a scenario where the user is performing continuous queries to the LBS provider. With the purpose to demonstrate this last vulnerability, we propose an algorithm of attack to identify the real location of the user in a query anonymized by the DLS and DLP algorithms.

Assuming that the LBS provider is not reliable, we propose a technique based on the selection of dummy locations whose fake locations are selected using as a criterion the Manhattan distance between locations in consecutive queries, the probability of each location occurrences, and the correlation between the response of previous requests and the current real user location. Also, our proposal protects not only the location information but also the content request, thus guaranteeing higher privacy to the users against attacks that exploit these vulnerabilities.

4. ATTACKING LBS PROVIDERS

Our attack algorithm is specific for requests made to the LBS provider that uses the dummy location technique. It seeks to identify the real location of the user by exploiting the acquired knowledge about the locations present in the request's location information. More precisely, it exploits the distance and correlation between the locations. Our attack algorithm uses the Manhattan distance metric to calculate the distance between the locations in consecutive queries sent by the same user to the LBS. To measure the correlation between the locations, our attack algorithm uses the Spearman's Rank Correlation Coefficient (SRCC) [Spearman 1904]. SRCC was chosen due to its robustness facing the limitations of the Pearson Correlation Coefficient, which is not able to handle non-linear correlations. It computes the similarity degree between two objects, comparing their attributes. In our work, we compare the locations' probability, coordinates, and class. Our attack, Algorithm 1, receives as input parameters:

(1) R: the current request.
(2) *limit*: refers to the maximum distance reachable by the user, calculated based on the formula $max_{\Delta s} = v \times \Delta t$, where v is an estimation of the average user speed and Δt is the time interval between the previous query and the current query.
(3) L: the list of locations covered by the LBS provider, and their respective side information, *i.e.*, its popularity, and the list of neighbors with their respective distance to the location.
(4) R': the last request sent to the service provider and its response.

The first step of our attack algorithm checks if there is a previous request R' sent by the user. If R' is empty (line 1), that is, there is no record of the user's previous request, the attack algorithm will identify as user's real location, the one $l \in R$ whose probability of being in a request made to the LBS is the highest of the locations in R.

If there is a previous request made by the user, which means R' is not empty (line 3), for each location $l \in R$, the algorithm will compute the manhattan distance between l and all locations $l' \in R'$. If there is at least one location l' where the distance $d_{l',l} <= limit$, the location l is a viable candidate for user's real location and is added to the set of reachable locations D (line 7). The other locations are discarded.

Algorithm 1: Attack

Input: R' , R, $limit$, L
Output: l_r

```
1  if R' == ∅ then
2  |  l_r ← Location l ∈ R with highest probability p_l;
3  else
4  |  foreach location l ∈ R do
5  |  |  foreach location l' ∈ R' do
6  |  |  |  if d_{l',l} <= limit then
7  |  |  |  |  D ← l;
8  |  |  |  end
9  |  |  end
10 |  end
11 |  Res ← locations present in the request's response R';
12 |  foreach location l ∈ D do
13 |  |  foreach location l' ∈ Res do
14 |  |  |  if c_{l',l} == Highest correlation found between the locations in D and Res then
15 |  |  |  |  C ← l;
16 |  |  |  end
17 |  |  end
18 |  end
19 |  l_r ← location l ∈ C with the highest probability p_l;
20 end
21 return l_r
```

Among the remaining locations in D, the attack verifies which one has the highest correlation with the locations in the response to the previous query R' (line 14). If there is more than one location with the highest correlation, it chooses as the real location the one with the highest popularity (line 19).

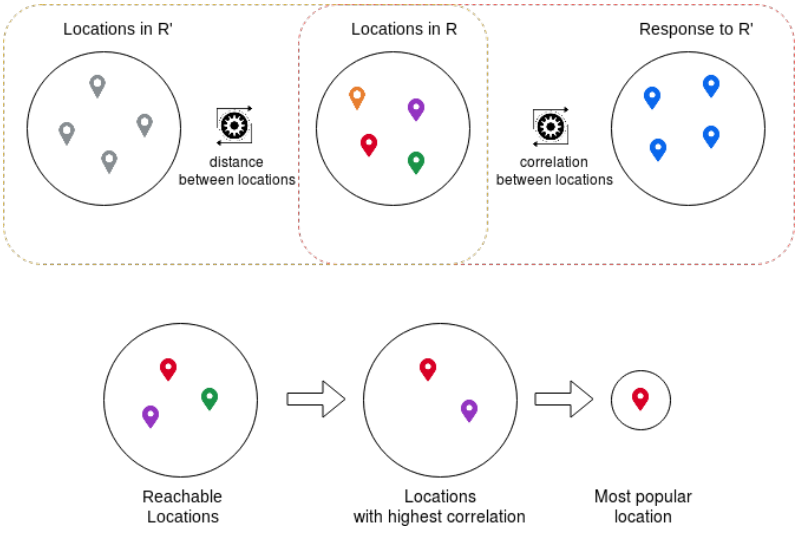

Fig. 3. Attack execution flow

Figure 3 shows the attack flow. First, it selects, among the request's locations, the reachable ones, removing the others. Second, it checks their correlations with the locations present in the response

to the previous request and selects the one with the highest correlation. If there is more than one location remaining, then it selects as guess to the real location, the one with the highest popularity.

5. PRIVLBS

In our LBS model, similar to the one described in Sun et al. [2017], the LBS provider is responsible for collecting the side information [Ma et al. 2013] for each location and to make it available to the user. The information contained in the side information will be used during the process of anonymization. A location l in our model represents a POI and it is defined by the triple $l = \{coordinates,\ label,\ popularity\}$. The side information collected in our model is restricted to the location's probability of each location in $\mathcal{L}$, the set of locations covered by the LBS.

PrivLBS is a privacy preservation algorithm that uses the dummy location selection technique, seeking to obfuscate the real location of the user among the other $k-1$ dummy locations present in the request. The central idea of the PrivLBS is to ensure that the LBS provider can not distinguish the user's real location from the other $k-1$ locations present in a request, even if the LBS provider has the knowledge about the user's previous request and the side information about the locations. However, PrivLBS does not only seek to protect user location information, but it also seeks to protect the contents of the request by generating a request containing up to k different request contents, increasing the user's privacy guarantee.

PrivLBS adopts three selection criteria during the process of anonymizing the requisition location information:

(1) Distance: as verified by the attack algorithm in section 4, the user's location is easily identified in a scenario of continuous queries where the request is anonymized using the dummy locations technique. Thus, the PrivLBS dummy location selection process only selects locations that are reachable by at least one location present in the user's previous request, *i.e.* locations where exists a possible displacement of the user from a location present in the previous query, and a location of the current query, taking into account user's speed and the time interval between the requests.
(2) Probability: PrivLBS, to avoid an attack that seeks to exploit the popularity of the locations present in the request, seeks to select locations that have probabilities of being in the request similar to the user's real location probability.
(3) Correlation: the last criterion is the correlation between each location present in the actual request and the locations present on the response for the user's previous query. We use the Spearman's Rank Correlation Coefficient (SRCC) to measure the correlation between two locations [Spearman 1904]. Since the correlation calculated by the SRCC indicates the similarity degree between two objects, a high correlation between a location present in a response and a location in a new request is an excellent guess about the real user's location. To avoid this scenario, PrivLBS selects locations with similar correlations.

PrivLBS combines these three criteria, distance, probability, and correlation, during the anonymization process to ensure the privacy of the real user location. However, it is still necessary to anonymize the request content, *i.e.*, the request for the location of a POI, based on the location information present in the query. The previous dummy localization models generates a request with a unique request content, and multiple locations ($req = \{(l_1, l_2, ..., l_k), request\ content\}$). Thus, since the request content is usually associated with a location class, it is exposed to inference attacks. As a way of protection, PrivLBS attempts to anonymize the class present in it, preventing the attacker from identifying the actual request content.

We can divide the PrivLBS anonymization process into three phases:

In the **first phase**, Figure 4, for each dummy location of the previous request sent to the service provider, subsets are constructed with locations that are within the maximum distance radius that the

user can travel in the time interval between the previous and the new query. If there are no previous requests, the dummy locations are chosen based on the location's popularity. In this case, it will be selected locations whose probability to be in a request is similar to the real location's probability. In this scenario, after the selection of dummy locations, PrivLBS follows directly to phase three of the anonymization process.

Fig. 4. PrivLBS first phase.

In the **second phase**, Figure 5, first, we identify the user's displacement behavior during the time interval between the last request and the current moment, to use this knowledge in the selection of dummy locations. For that purpose, using the Spearman's rank coefficient, we measure the correlation between the response to the real location present in the previous request and the current user's location, denoted as c_{real}. In the first phase, for each dummy location l' of the previous request, a set with reachable locations from it were generated. PrivLBS selects from each of these sets a dummy location l based on its correlation with the location present on the response to the location l' that generated the set. Thus, the location l whose correlation with the response to l' is closest to c_{real} will be chosen. In the end, PrivLBS selects $k-1$ dummy locations that are reachable by at least one location of the previous query and produces a displacement behavior with the previous dummy locations, similar to the behavior between the previous user's real location and the current user's real location of the user.

In the **third phase**, Figure 6, we will anonymize the content request. For each dummy location selected in the new request, a class is defined to be associated with the content of the request that will be appended to this dummy location. The class selection will respect the class distribution in the data set. Thus, if 5% of the locations in the data set are from the "Restaurant" class, this class will have the probability of being chosen to be associated with a request of 5%.

Figure 7 represents the complete flow of the PrivLBS anonymization process. We can observe each one of the three phases of PrivLBS. In phase 1, the criterion of the distance between the locations is treated. In the second phase, the correlation and popularity are handled. Finally, in the third phase, the anonymization of the request content is treated.

The PrivLBS Algorithm (2) receives as input parameters:

(1) k: the degree of privacy defined by the user. It is the number of locations that will be present in the request, which therefore also defines the maximum desired probability of re-identification of the actual location.

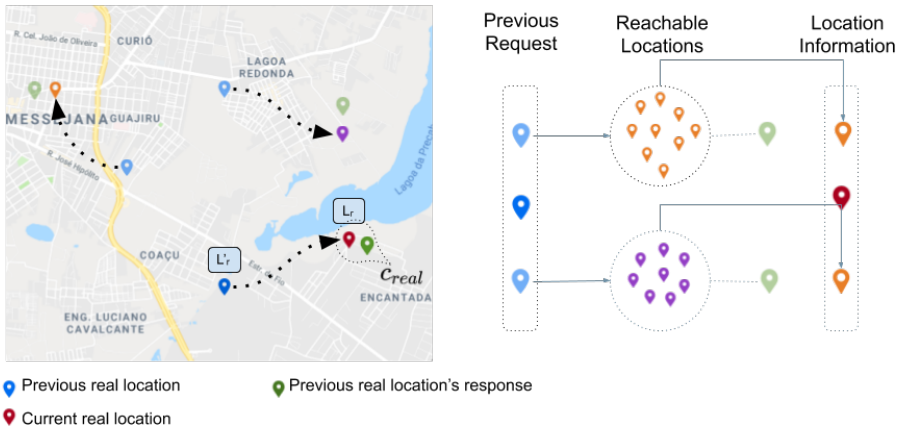

Fig. 5. PrivLBS second phase.

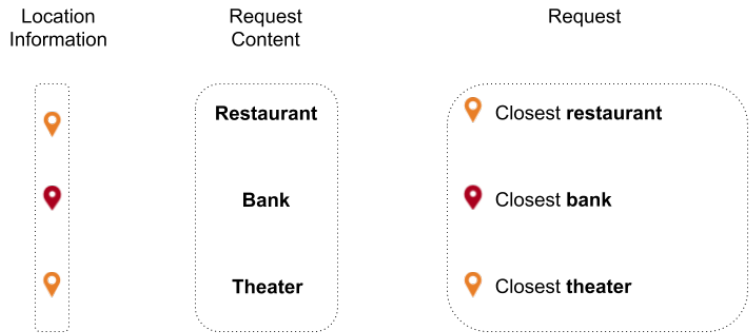

Fig. 6. PrivLBS third phase.

(2) L: the list of locations covered by the LBS provider, and its complementary information, *i.e.* popularity, class, and list of neighbors with its respective distances to the location.

(3) l_r: real user location, with $l_r \in L$.

(4) R': last request sent to the service provider and its response.

The user is responsible for storing the history of the last requests sent to the LBS and their responses. If the user history R' is clear, which indicates that the user is making his first request, the parameter R' is empty, Algorithm 2. In the case, PrivLBS selects $k-1$ locations whose probabilities of being in a request are similar to the real location, measured by their popularity. Thus, the set C (line 3) is created, containing $2k$ locations whose probabilities are closest to l_r. We define this size of $2k$ to ensure a representation of locations that allows a safe random choice against attacks that seek to identify the real location based on the variance of the probabilities of the locations present in the request. From C, $k-1$ locations will be randomly chosen. To avoid selecting locations far away from others, which would eventually compromise the user's privacy, we adopt the isolation rate. It measures how isolated a location is. The Equation 4 (line 5) calculates the isolation rate, which we measure the ratio between the number of neighbors of a location and the minimum number of neighbors we

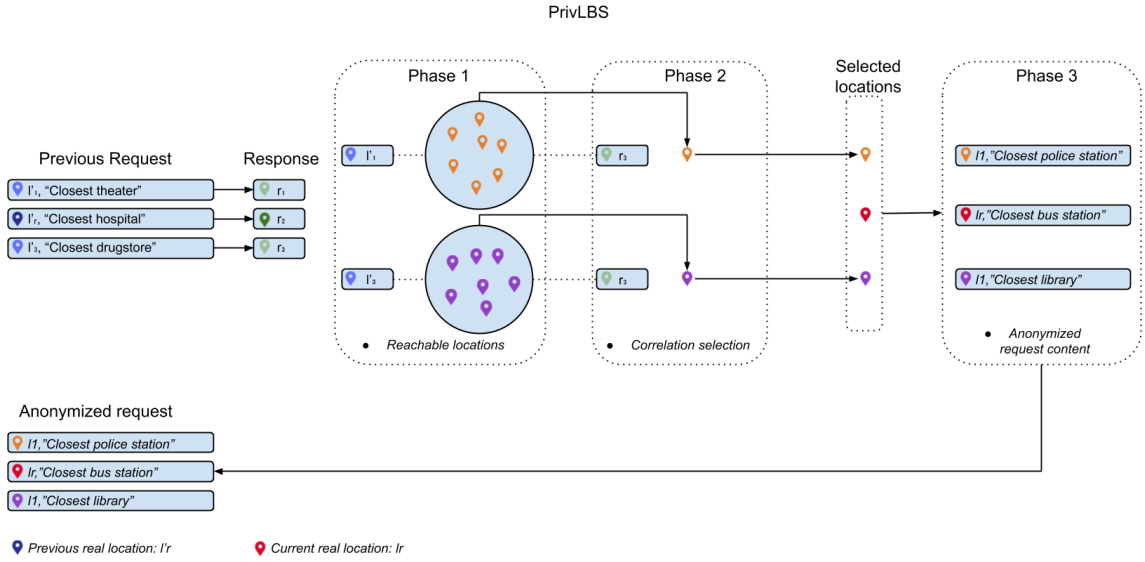

Fig. 7. PrivLBS

judge necessary to reach a certain degree of unpredictability in the choice of the dummy locations. Once again, we have defined this minimum amount of $2k$ to guarantee representation of the locations that allow the right choice in terms of privacy protection. This value has guaranteed a high degree of privacy in our experiments. Thus, locations whose isolation rate is greater than or equal to 1 will be chosen, in which case the number of neighbors of the location is equal to $2k$, or higher than the isolation rate of the real location (line 6).

$$isolation\ rate = \frac{number\ of\ reachable\ neighbors}{2k} \tag{4}$$

If R' is not empty (line 13), which indicates that there is an earlier query sent to the LBS, it is necessary to have this previous request as a reference in the choice of the dummy locations. The first step is to calculate the reachability area for each of the previous query locations. Given the average speed of the user v in the time interval between two requests $\Delta t = t - t'$, with $t > t'$, we define the maximum reachable distance as $max_{\Delta s} = v \times \Delta t$ (line 15). Thus, for each dummy location $l' \in R'$ we define the set of reachable locations D (line 16), denoting the set of locations $l \in L$ whose Manhattan distance of l', denoted by $d_{l',l}$, is not greater than $max_{\Delta s}$. Importantly, we make it clear that we adopt the Manhattan distance since it guarantees a more accurate distance measure in a real scenario than the Euclidean distance, although PrivLBS can use any distance criterion.

To ensure greater representativeness of the locations in the sets D, we define the minimum size of $2k$ locations by adding the nearest locations that are outside the reachability area until the minimum size is reached. This action decreases the assurance that all locations will be reachable by at least one location of the previous request. However, it is critical to ensure a minimum number of locations that allow an appropriate choice in the next stages of the selection process. Experiments have shown that the impact of this measure has not generated significant privacy loss.

From each set of reachable locations D, a dummy location is selected to compose the request. This selection has as criterion, the correlation between the locations of the set of reachable locations D and the response to the location l' from which D was generated. To compute the correlation between two locations l_i e $l_j \in L$ we use the coefficient of Spearman, defined by $c_{l_i,l_j} \in [-1, 1]$. The goal is

Algorithm 2: PrivLBS

Input: k, L, l_r, R'
Output: R

1 **if** $R' == \emptyset$ **then**
2 $p_r \leftarrow$ *probability of the real location being in the requisition*;
3 $ti_r \leftarrow$ *isolation rate of* l_r;
4 $C \leftarrow$*select the 2k locations* $l_i \in L$ *whose probability* p_{l_i} *are the closest to* p_r;
5 **for** $i = 0 \rightarrow k$ **do**
6 **if** *Exist* $l \in C$, *with* $ti_l \geq ti_r$ *or* $ti_l \geq 2k$ **then**
7 $loc_{info} \leftarrow l$;
8 **else**
9 $loc_{info} \leftarrow$ *randomly selected a location* $l \in C$;
10 **end**
11 $C \leftarrow C - l$;
12 **end**
13 **else**
14 $max_{\Delta s} \leftarrow v \times \Delta t$;
15 $c_{l_r,r_{l'_r}} \leftarrow$ *Correlation between the current user's real location and the response to the previous' user real location*;
16 **foreach** $l' \in R'$ **do**
17 $D \leftarrow$ *From the neighborhood of* l' *select all locations* l *where* $d_{l',l} <= max_{\Delta s}$;
18 $loc_{info} \leftarrow$ select $l \in D$ *whose* $c_{l,r_{l'}}$ *is the closest to* $c_{l_r,r_{l'_r}}$;
19 **end**
20 **end**
21 **foreach** $l \in loc_{info}$ **do**
22 $cl \leftarrow$ *select a class present in L with a probability function according to the data set distribution*;
23 $req \leftarrow$ *generate a request content with the class* cl;
24 $R \leftarrow$ *add the tuple (l,req) to the request* R;
25 **end**
26 **return** R

to capture the behavior of the moving user regarding to his or her real locations in the requests, and reproduce this behavior in the selection of the dummy locations. In this process, we first calculate the correlation between the current location of the user l_r and the response to the real location in the previous query $r_{l'_r}$ (line 17), denoted by $c_{l_r,r_{l'_r}}$. From this point, for each set D generated from a dummy location $l' \in R'$, we select the location $l \in D$ whose $c_{l,r_{l'}}$ is the closest to $c_{l_r,r_{l'_r}}$ (line 18).

Finally, for each dummy location selected, a request content associated with a class is assigned (line 23), respecting the class distribution present in the locations in L. In this way, the contents of the request are also anonymized, making it challenging to discover the real location and preventing inferences from being made about them.

As output from the PrivLBS algorithm, we have a request

$$req = \{(l_1, content_1), (l_2, content_2), ...(l_k, content_k)\}$$

(line 26), where the LBS provider sees all locations with similar characteristics, inducing the probability of recognition of the real location at $\frac{1}{k}$. It is important to note that during the process of anonymization, in a few steps, we define a minimum amount of $2k$ locations, for instance, in the construction of the set C, or the number of neighbors. This value was defined empirically. Our experiments have shown that with this value, PrivLBS achieves the required level of privacy.

6. EXPERIMENTAL RESULTS

For evaluating the performance of PrivLBS in comparison with the algorithms DLS [Niu et al. 2015] and DLP [Sun et al. 2017], we have conducted a series of experiments. We analyze the recognition rate of the real location under the ADLS algorithm proposed in [Sun et al. 2017] and our attack. The ADLS aims to identify the real location in an anonymized request by a dummy location selection, exploiting the locations' probability. Also, we measure the entropy of the requests generated by the anonymization algorithms, which consists of the uncertainty of identifying the actual location among the selected dummy locations [Serjantov and Danezis 2003], computed through the Equation 2, where q_i stands for the normalized location's l_i popularity. Finally, we verify the identification rate of the real class present in the contents of the requests based on the probability of each class being present in the request.

We use a real data set available by Chicago Transit Authority[1] (CTA). It contains 11,593 bus stations, their locations, with information about latitude, longitude, and average daily boarding. For our experiments, we add a new field, the location's class. We estimated 90 classes following the number of location categories present in the Google API[2]. The distribution of the classes in the dataset followed a normal distribution.

We set up our experiments with ten thousand users moving and submitting continuous queries to the LBS provider. Each user performs four queries. The time interval between each query ranges from 30 seconds to 10 minutes. User speed ranges from 5 kilometers per hour to 80 kilometers per hour. Besides, we assume that the user has a probability between 10% and 20% to move in the direction to the location present in the previous query response, reaching it if the time interval and user speed are sufficient to that. After each request is anonymized by the algorithms, we compute its entropy and apply the attack algorithms over it to measure the recognition rate of the real location present in the request.

	Degree of anonymization k				
	2	**4**	**8**	**12**	**16**
DLS	0,691	1,386	2,079	2,484	2,771
DLP	0,693	1,389	2,079	2,484	2,772
PrivLBS	0,68	1,375	2,066	2,470	2,758

Table I. Entropy based on the location's probability.

Table I shows the entropy of the requests based on the location's popularity. It indicates the degree of uncertainty about the locations present in the requisition. The higher the entropy is, the greater is the degree of similarity between the locations present in the request. The maximum possible entropy, $\log_2 k$, is obtained when all the locations present in the request have the same probability of being present on a request. We can observe the DLS and DLP algorithms reach an entropy slightly larger than PrivLBS, which is expected since they use the entropy as a cost function during the dummy location selection, differently from our approach that adopts another selection criterion besides the probability. Therefore, PrivLBS, despite looking to select locations with similar probabilities, eventually narrows the search domain by requiring reachable locations. However, the loss of entropy is not significant, presenting values close to any degree of privacy measured.

This behavior ends up being reflected in the recognition rate of the real location. Figure 8 shows that for any degree of privacy k, the PrivLBS recognition rate was always less than $\frac{1}{k}$, ensuring the user's privacy. We can observe that for $k = 8$, both PrivLBS and DLP had similar recognition rate, different from the DLS that presented the worst recognition rate of the three algorithms.

[1] http://www.transitchicago.com
[2] https://developers.google.com/location-context/

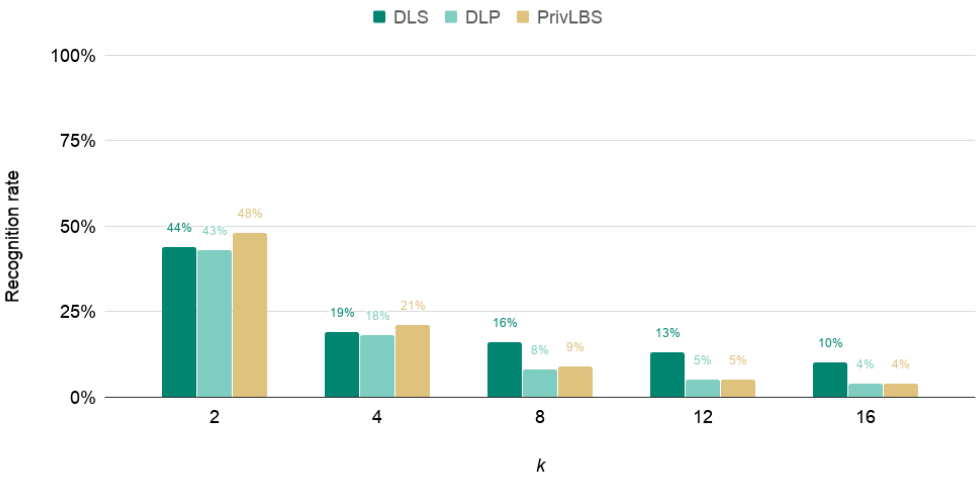

Fig. 8. Real location recognition rate when applied the ADLS attack algorithm

Table II shows the entropy of the requests based on the location's popularity, distance, and correlation. For any degree of privacy k, the entropy of the request anonymized by the PrivLBS is the largest of the methods compared, demonstrating that the locations selected by our algorithm have a higher degree of similarity, generating greater indistinguishability between the locations present in the request.

	Degree of anonymization k				
	2	**4**	**8**	**12**	**16**
DLS	0,613	1,294	1,986	2,392	2,680
DLP	0,615	1,294	1,986	2,392	2,680
PrivLBS	0,652	1,325	2,014	2,401	2,710

Table II. Entropy based on location's probability, distance, and correlation

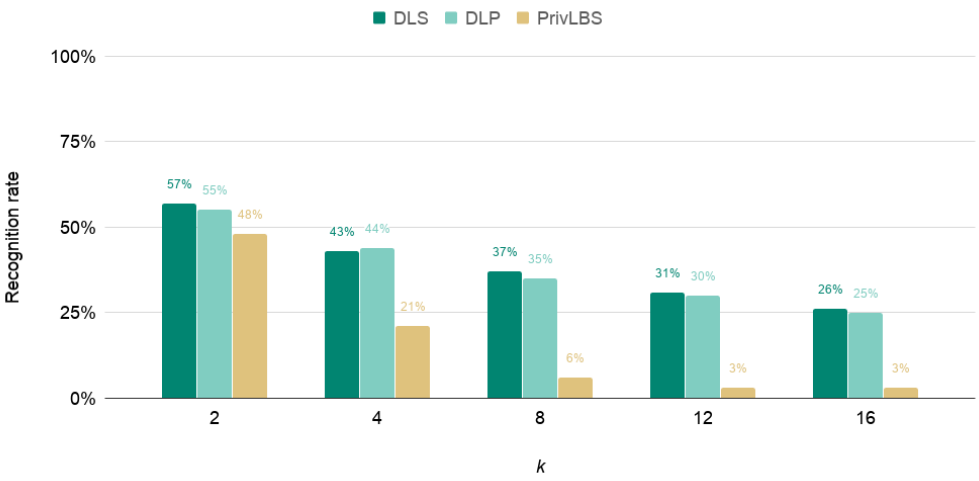

Fig. 9. Real location recognition rate when applied our attack algorithm.

Once the analysis of the request's entropy was made, we proceed to analyze the recognition rate when applied our proposed attack algorithm. For this experiment, in our attack algorithm, we estimate a user moving at 40 kilometers per hour, which is used to calculate the maximum distance that a user may go through between two consecutive requests. This value was set since it is in the range of the moving users in our experiments. Once again, we calculate the recognition rate of the real location present in a request for different degrees of privacy k. Figure 9 shows the fraction of locations recognized as the real user location by our attack algorithm. The requests anonymized by the DLS and DLP have a high recognition rate under our attack, not ensuring the user's privacy. On the other hand, the requests generated by the PrivLBS had a recognition rate always lower than $\frac{1}{k}$.

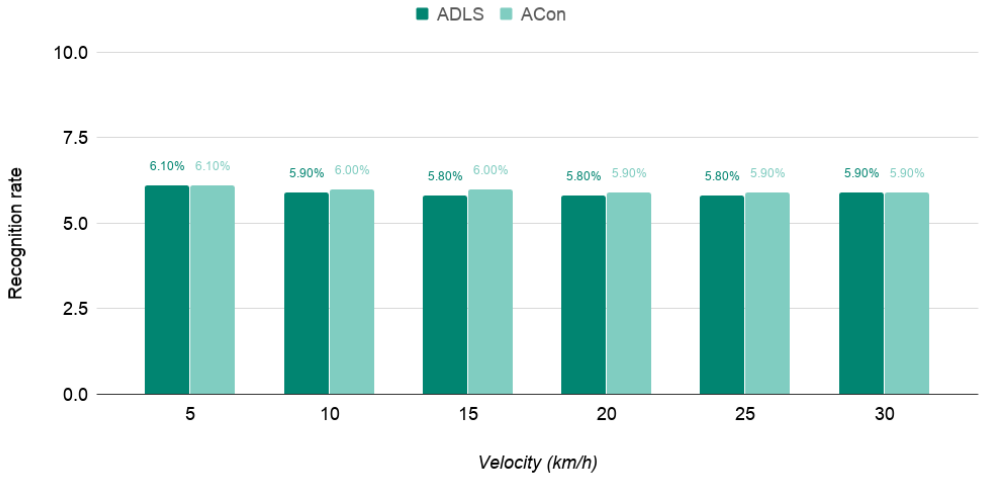

Fig. 10. Real location recognition rate when applied the attack algorithms for different user velocities.

Since PrivLBS has a more rigorous selection process, where the locations need to be at a certain distance from the locations present in the previous request, we found the need to evaluate the behavior of the anonymous requests varying the user speed. The Figure 10 shows the recognition rate of the user's real location anonymized by PrivLBS under attack by the ADLS and our attack algorithm, in a scenario where the user speed is low, which could result in a small set of reachable locations. We can observe that, for $k = 8$, the real location identification rate remained constant, close to 6%, as the speed increased, demonstrating that PrivLBS can protect user location information even in an adverse scenario with few reachable locations.

Finally, we analyze the protection of the request content. We apply an attack on the content to identify the class associated with the user's real location, pointing as the real class the one with the greatest probability of being present in the content of the request among all classes present in the request content. The Figure 11 shows the class identification rate in the content anonymized by PrivLBS. The results demonstrate that PrivLBS can efficiently protect the class present in the real request content, obtaining an identification rate close to $\frac{1}{k}$ for any degree of privacy k, different from the others approaches that do not perform any anonymization on the request content.

7. CONCLUSION

In this article, we presented PrivLBS, an approach to preserve the user's privacy in LBS. PrivLBS protects not only the location information but also the request content. It selects $k-1$ dummy locations considering the location's popularity, correlation, and the Manhattan distance between two locations to compose the location information. For each dummy location, a location class is selected to anonymize

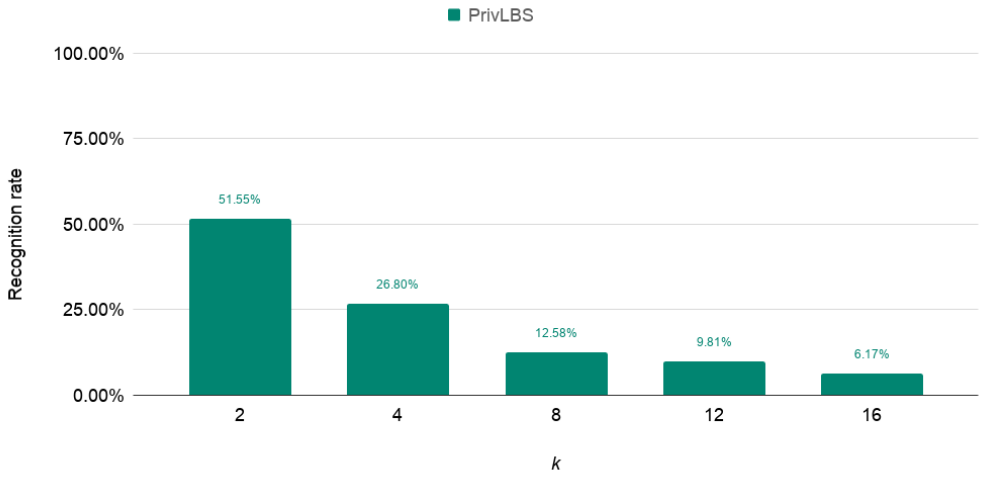

Fig. 11. Attack on the request content anonymized by PrivLBS.

the request content. Additionally, we proposed an attack algorithm capable of identifying the real user's location in an anonymized request by the dummy location selection technique. Experimental results show that PrivLBS has a lower probability of revealing the user's real location in comparison with the DLS and DLP algorithm. Also, we demonstrate that the real user's request content in a query made by PrivLBS has its class content protect among the other $k-1$ classes present in the request.

As limitations, PrivLBS generates an overhead due to the cost of the dummy location selection, which increases with the number of locations present in the request. Also, the anonymization process depends on the number of reachable locations. Although we introduce some measures to turn around this limitation, a study with other datasets would be necessary to confirm PrivLBS performance. As future work, we intend to explore some other techniques, such as Differential Privacy, capable of guaranteeing the user's privacy independently of the attacker's knowledge, minimizing the loss of data utility, and consequently ensuring the quality of the location-based service. We want to conduct other experiments, with a greater variety of scenarios, aiming to identify a solution that best fits.

REFERENCES

Andrés, M. E., Bordenabe, N. E., Chatzikokolakis, K., and Palamidessi, C. Geo-indistinguishability: Differential privacy for location-based systems. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*. New York, NY, USA, pp. 901–914, 2013.

Bamba, B., Liu, L., Pesti, P., and Wang, T. Supporting Anonymous Location Queries in Mobile Environments with Privacygrid. In *Proceedings of the 17th International Conference on World Wide Web*. New York, NY, USA, pp. 237–246, 2008.

Brito, F. T., Neto, A. C. A., Costa, C. F., Mendonça, A. L., and Machado, J. C. A Distributed Approach for Privacy Preservation in the Publication of Trajectory Data. In *Proceedings of the 2nd Workshop on Privacy in Geographic Information Collection and Analysis*. New York, NY, USA, pp. 5:1–5:8, 2015.

Dewri, R., Ray, I., Ray, I., and Whitley, D. On the Optimal Selection of k in the k-Anonymity Problem. In *24th ICDE International Conference on Data Engineering*. Cancun,Mexico, pp. 1364–1366, 2008.

Duckham, M. and Kulik, L. A Formal Model of Obfuscation and Negotiation for Location Privacy. In *Pervasive Computing*. Berlin, Heidelberg, pp. 152–170, 2005.

Dwork, C. Differential Privacy. In *33rd International Colloquium on Automata, Languages and Programming*. Venice, Italy, pp. 1–12, 2006.

Gedik, B. and Liu, L. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *IEEE Transactions on Mobile Computing* 7 (1): 1–18, 2008.

Ghinita, G. Privacy for Location-based Services. *Synthesis Lectures on Information Security, Privacy, and Trust* 4 (1): 1–85, 2013.

Griffith, D. and Chun, Y. Spatial Autocorrelation and Spatial Filtering. In M. M. Fischer and P. Nijkamp (Eds.), *Handbook of Regional Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 1477–1507, 2014.

Hu, H., Chen, Q., and Xu, J. VERDICT: privacy-preserving authentication of range queries in location-based services. In *2013 IEEE 29th International Conference on Data Engineering ICDE*. Brisbane, QLD, Australia, pp. 1312–1315, 2013.

Hubaux, J., Theodorakopoulos, G., Boudec, J. L., and Shokri, R. Quantifying Location Privacy. In *2011 IEEE Symposium on Security and Privacy(SP)*. Oakland, California, USA, pp. 247–262, 2011.

Kido, H., Yanagisawa, Y., and Satoh, T. An anonymous communication technique using dummies for location-based services. In *ICPS '05. Proceedings. International Conference on Pervasive Services, 2005.* Santorini, Greece, pp. 88–97, 2005.

Liu, B., Zhou, W., Zhu, T., Gao, L., and Xiang, Y. Location Privacy and Its Applications: A systematic study. *IEEE Access* vol. 6, pp. 17606–17624, 2018.

Lu, R., Lin, X., Shi, Z., and Shao, J. PLAM: A privacy-preserving framework for local-area mobile social networks. In *INFOCOM, 2014 Proceedings IEEE*. Toronto, ON, Canada, pp. 763–771, 2014.

Ma, C. Y. T., Yau, D. K. Y., Yip, N. K., and Rao, N. S. V. Privacy Vulnerability of Published Anonymous Mobility Traces. *IEEE/ACM Trans. Netw.* 21 (3): 720–733, 2013.

Machanavajjhala, A., Gehrke, J., Kifer, D., and Venkitasubramaniam, M. L-diversity: privacy beyond k-anonymity. In *22nd International Conference on Data Engineering (ICDE'06)*. Atlanta, GA, USA, pp. 24–24, 2006.

Meyerson, A. and Williams, R. On the Complexity of Optimal K-Anonymity. In *Proceedings of the 23rd ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. Paris, France, pp. 223–228, 2004.

Neto, E. R. D., Mendonça, A. L. C., Brito, F. T., and Machado, J. C. PrivLBS: uma abordagem para preservação de privacidade de dados em serviços baseados em localização. In *Brazilian Symposium on Databases SBBD*. Rio de janeiro, Brazil, pp. 109–120, 2018.

Niu, B., Gao, S., Li, F., Li, H., and Lu, Z. Protection of location privacy in continuous LBSs against adversaries with background information. In *2016 International Conference on Computing, Networking and Communications (ICNC)*. Kauai, HI, USA, pp. 1–6, 2016.

Niu, B., Li, Q., Zhu, X., Cao, G., and Li, H. Achieving k-anonymity in privacy-aware location-based services. In *INFOCOM, 2014 Proceedings IEEE*. Toronto, ON, Canada, pp. 754–762, 2014.

Niu, B., Li, Q., Zhu, X., Cao, G., and Li, H. Enhancing privacy through caching in location-based services. In *Computer Communications (INFOCOM), 2015 IEEE Conference on*. Kowloon, Hong Kong, pp. 1017–1025, 2015.

Serjantov, A. and Danezis, G. Towards an Information Theoretic Metric for Anonymity. In *Privacy Enhancing Technologies*. Springer, Berlin, Heidelberg, pp. 41–53, 2003.

Spearman, C. The proof and measurement of association between two things. *The American journal of psychology* 15 (1): 72–101, 1904.

Sun, G., Chang, V., Ramachandran, M., Sun, Z., Li, G., Yu, H., and Liao, D. Efficient location privacy algorithm for Internet of Things (IoT) services and applications. *Journal of Network and Computer Applications* vol. 89, pp. 3–13, 2017.

Sun, G., Liao, D., Li, H., Yu, H., and Chang, V. L2P2: A location-label based approach for privacy preserving in lbs. *Future Generation Computer Systems* vol. 74, pp. 375–384, 2017.

Tsoukaneri, G., Theodorakopoulos, G., Leather, H., and Marina, M. K. On the Inference of User Paths from Anonymized Mobility Data. In *2016 IEEE European Symposium on Security and Privacy (EuroS P)*. Saarbrucken, Germany, pp. 199–213, 2016.

Ullah, I. and Shah, M. A. A novel model for preserving Location Privacy in Internet of Things. In *2016 22nd International Conference on Automation and Computing (ICAC)*. Colchester, UK, pp. 542–547, 2016.

Vu, K., Zheng, R., and Gao, J. Efficient algorithms for K-anonymous location privacy in participatory sensing. In *2012 Proceedings IEEE INFOCOM*. Orlando, FL, USA, pp. 2399–2407, 2012.

Wang, L., Yang, D., Han, X., Wang, T., Zhang, D., and Ma, X. Location privacy-preserving task allocation for mobile crowdsensing with differential geo-obfuscation. In *Proceedings of the 26th International Conference on World Wide Web*. Perth, Australia, pp. 627–636, 2017.

Ying, B. and Makrakis, D. Protecting location privacy with clustering anonymization in vehicular networks. In *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on*. IEEE, Toronto, ON, Canada, pp. 305–310, 2014.