ResPred: A Privacy Preserving Location Prediction System Ensuring Location-based Service Utility

Arielle Moro and Benoît Garbinato
Institute of Information Systems, University of Lausanne, Lausanne, Switzerland

Keywords: Location Prediction, Location Privacy-preserving Mechanism, Threat Model, Inference Attack, Location-based Services.

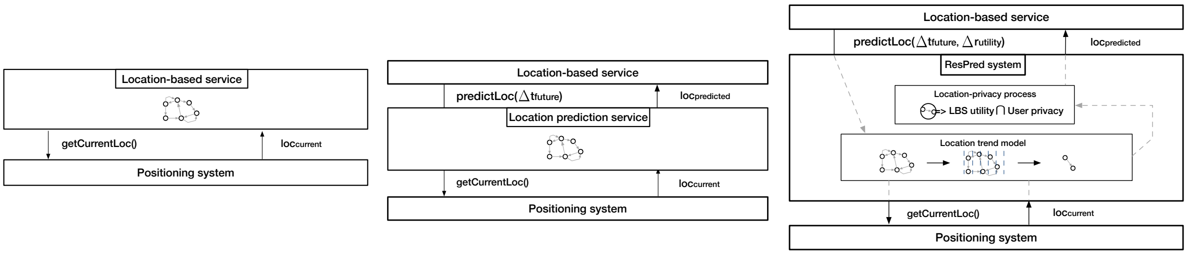
Abstract: Location prediction and location privacy has retained a lot of attention recent years. Predicting locations is the next step of Location-Based Services (LBS) because it provides information not only based on where you are but where you will be. However, obtaining information from LBS has a price for the user because she must share all her locations with the service that builds a predictive model, resulting in a loss of privacy. In this paper we propose *ResPred*, a system that allows LBS to request location prediction about the user. The system includes a location prediction component containing a statistical location trend model and a location privacy component aiming at blurring the predicted locations by finding an appropriate tradeoff between LBS utility and user privacy, the latter being expressed as a maximum percentage of utility loss. We evaluate *ResPred* from a utility/privacy perspective by comparing our privacy mechanism with existing techniques by using real user locations. The location privacy is evaluated with an entropy-based confusion metric of an adversary during a location inference attack. The results show that our mechanism provides the best utility/privacy tradeoff and a location prediction accuracy of 60% in average for our model.

1 INTRODUCTION

In recent years, predicting future locations of users has become an attractive topic for both the research community and companies. Location prediction can boost the creation of new Location-Based Services (LBS) in order to help users in their daily activities. For example, a LBS could send personalized information to users, such as the menu of different restaurants the users could like in the vicinity of a location in which they will probably be at a specific time, e.g., Monday between 11:30 and 12:00 am. In order to obtain future locations of a user, a LBS needs to build a predictive model containing spatial and temporal information. However, this leads to a first location privacy issue because the user must send all her raw locations to a third-party entity as described in Figure 1 (a). In this architecture, the LBS, which can be malicious, is installed on the mobile device of the user and gathers all user locations. To preserve location privacy, the idea is to create a location predictive model in a trusted component that can be stored at the operating system level of the mobile device. In this context, the trusted component itself will provide the future locations of the user to the LBS as depicted in Figure 1 (b). Even after a large number of requests performed by the LBS, it should not be able to reconstruct the entire predictive model of the user but may have a good partial view of her model. As a result, this is a undeniable second location privacy issue. It has been demonstrated in the literature that sharing accurate locations has a real cost for a user because a potential adversary cannot only discover a lot of sensitive information related to the user but also identify her by just performing simple location attacks as described by Krumm in (Krumm, 2007). In addition, the authors of (Zang and Bolot, 2011) show that a few number of user's locations only might highly compromise the location privacy of a user.

Because of the availability of different positioning systems on mobile devices, LBS are very convenient for daily activities. Consequently, users cannot completely avoid using LBS. However, users must know that it is fundamental to preserve their privacy when they are using LBS. Currently, users can only enable or disable the access to locations for specific applications and sometimes reduce the precision of the locations obtained with a positioning system. These options depend on the operating system itself. These simple choices are not adapted to the context of our work because we want to preserve the location privacy of the user at a higher level, which is a loca-

Moro, A. and Garbinato, B.
ResPred: A Privacy Preserving Location Prediction System Ensuring Location-based Service Utility.
DOI: 10.5220/0006710201070118
In *Proceedings of the 4th International Conference on Geographical Information Systems Theory, Applications and Management (GISTAM 2018)*, pages 107-118
ISBN: 978-989-758-294-3



(a) First architecture

(b) Second architecture

(c) Third architecture including ResPred system

Figure 1: Problem/contribution overview through three different system architectures.

tion prediction level. In order to protect raw locations of a user, some existing Location Privacy Preserving Mechanisms (LPPMs) can be applied, such as spatial perturbation, spatial cloaking, sending dummy locations as well as spatial rounding, as discussed in (Krumm, 2007; Gambs et al., 2011; Agrawal and Srikant, 2000; Gruteser and Grunwald, 2003; Kido et al., 2005). Nevertheless, these mechanisms may quickly decrease the utility level of a LBS as the level of protection increases, up to the point when the LBS becomes unusable.

In this paper, we present a privacy preserving location prediction system called *ResPred*, *res* and *pred* mean respect (i.e., respect the privacy of users) and prediction respectively. This system allows LBS to request future location of users. For instance, a LBS can display information containing future public transportation departures located in the vicinity of the predicted location returned by *ResPred* on the mobile device of the user in advance. Figure 1 (c) presents the *ResPred* system that contains two components: one component focuses on the location prediction and the second on the location privacy. We assume that the *ResPred* system is created at the operating system level of the mobile device and that the *ResPred* system and the positioning system are trusted. The system includes a location prediction component based on a statistical location trend model and a location privacy component helping to blur the predicted locations by finding an appropriate tradeoff between the LBS utility and the user privacy preference expressed as a maximum percentage of utility loss. We also assume that the LBS is untrusted, which indicates that it is a possible adversary. As depicted in Figure 1 (c), the LBS requests the future location of the user by indicating a time duration between the current time and the time of the desired predicted location and the system returns a predicted location that will be found by exploring the location trend model and protected by our LPPM. The predicted location is more specifically transformed according to the required utility level of the LBS and the maximum utility level that the user is willing to sacrifice in order to protect her location privacy. We evaluate our system from a utility/privacy perspective, which is the crucial aspect of our approach. In addition, we compute the location prediction accuracy of the location trend model. We chose real mobility traces coming from two datasets, the*PrivaMov* dataset described in (Ben Mokhtar et al., 2017) and a private dataset collected by a researcher in Switzerland. The first part of the utility/privacy evaluation consists in assessing the utility level of our LPPM and two other well-known mechanisms described in the literature, namely the rounding and the Gaussian perturbation. The second part of the utility/privacy evaluation focuses on the measurement of the confusion level of an adversary performing a location attack on the received predicted locations from the *ResPred* system. The metric used to evaluate this confusion level is based on the well-known Shannon entropy. The results show that our location privacy preserving mechanism provides the best utility/privacy tradeoff compared to the other evaluated mechanisms as well as a good location prediction accuracy for the analyzed users. The contributions of this paper are listed below.

- We describe a system, called *ResPred*, allowing LBS to request future location of a user.
- We present a statistical model containing location trends of a user per time slice, helping to extract short, mid and long-term predicted locations.
- We describe a LPPM enabling to reach an appropriate utility/privacy tradeoff.
- We use real user locations to assess our system and, more specifically, its two components.

The paper is organized as follows: in Section 2 we begin with the description of the system model containing the formal definitions used in the paper. Section 3 presents the problem addressed in this paper, while the *ResPred* system is described in Section 4. Then, we present the evaluation of the system from a utility/privacy perspective in Section 5. In addition, we also evaluate the location prediction accuracy of the location trend model of the *ResPred* system. We detail the closest work to the two main subjects of this paper in Section 6, which are the location privacy as well as the location prediction. Finally, we highlight the most important findings of the paper and discuss future work in Section 7.

2 SYSTEM MODEL

This section focuses on describing the key definitions used to present our system. In order to facilitate the analysis of locations of a user, the time is discretized. We also introduce Regions Of Interest (ROIs) on which the location predictive model is based. Finally, we present the threat model that describes the context used to evaluate the location privacy.

2.1 User and Locations

We consider that a user is moving on a geodesic space and is owning a mobile device that is able to detect her locations as well as when they are captured via a positioning system, e.g., GPS, WiFi or radio cells. A location is described as a triplet $loc = (\phi, \lambda, t)$ where ϕ and λ are the latitude and longitude of the location in the geodesic space, and t is the time when the location was obtained from the positioning system. Locations are formally represented as a sequence $L = \langle loc_1, loc_2, \cdots, loc_n \rangle$. A subsequence of successive location of L is described as follows $l_{sub_i} = \langle loc_1, loc_2, \cdots, loc_m \rangle$ in which the first location of this subsequence is noted $l_{sub_i}.loc_{first}$ and the last location is $l_{sub_i}.loc_{last}$. We can express the latitude, longitude and time of a location loc_i by directly writing $loc_i.\phi$, $loc_i.\lambda$ and $loc_i.t$ respectively.

2.2 Temporal Discretization

In order to discretize time, we compute n slices generated according to the chosen temporal granularity and time span, e.g., every 20 minutes during one week. A time slice is a triplet defined as follows $ts = (t_{starting}, t_{ending}, index)$ where $t_{starting}$ (Monday - 7:00 am) and t_{ending} (Monday - 7:20 am) represent the starting time and ending time of the time slice and $index$ is its unique identifier ranging between 1 and n (n represents the total number of computed time slices). For instance, if we generate all time slices having a duration of 20 minutes during a period of 1 week, we will obtain 504 time slices. All the possible time slices are represented as a sequence called $timeslices$, such that $timeslices = \langle ts_1, ts_2, \cdots, ts_n \rangle$. In addition, we introduce a function called $convert(\langle loc_1, loc_2, \cdots, loc_m \rangle)$ translating a sequence of one or several successive locations into a sequence of one or several successive time slices called $timesliceTab$, m being the total number of location(s) to convert. This sequence is described as follows: $timesliceTab = \langle ts_1, ts_2, \cdots, ts_n \rangle$ in which n is the total number of successive time slices.

2.3 Regions of Interest

A region of interest (ROI) is defined as a circular area visited by a user during a certain period of time, which is a quadruplet of the form $roi = (\phi, \lambda, \Delta r, visits)$. Items ϕ and λ are the coordinates of the center of the ROI in a geodesic space. Δr is the radius of the ROI and $visits$ is a sequence of subsequences of L such as $visits = \langle l_{sub_1}, l_{sub_2}, \cdots, l_{sub_m} \rangle$ in which each subsequence of successive locations is contained in L such that $\forall l_{sub_i} \in visits, l_{sub_i} \subset L$ and $l_{sub_i}.loc_{last}.t < l_{sub_{i+1}}.loc_{first}.t$. Each visit of a ROI has a duration equal or greater than a threshold, called Δt_{min}, such as $\forall l_{sub_i} \in visits, l_{sub_i}.loc_m.t - l_{sub_i}.loc_1.t >= \Delta t_{min}$. In addition, all locations of the visits are contained in the ROI spatially described by the first three items of it, i.e., latitude, longitude and radius. The set containing all ROIs of a user can be noted as follows: $rois = \{roi_1, roi_2, \cdots, roi_n\}$. The last and important characteristic of the ROI is that there is no spatial intersection between ROIs. This means that, if two ROI candidates intersect during the discovery process of ROIs, they will be merged and a new ROI is created from these two ROI candidates.

2.4 Threat Model

We consider a threat model that takes into account a honest but curious adversary in the form of a LBS using *ResPred*. The LBS will try to infer future locations of the user based on a location history gathered by requesting *ResPred*. This location history contains all predicted locations sent by *ResPred* and consists in its unique background knowledge on which the location attack will be performed. This history is not complete because we consider that the LBS will not request *ResPred* constantly but a limited number of times in a random manner during a certain time slice or by following the usual use of the LBS by the user, e.g., everyday at the end of the afternoon. The honest but curious behavior of the LBS also means that it will not try to break the sharing protocol or obtain the location predictive model of the system *ResPred*. In addition, we consider that the LBS always gives adapted parameters to its service to the *ResPred* system, more specifically the values of the parameters Δt_{future} and $\Delta r_{utility}$ as depicted in Figure 1 (c) or Figure 2.

3 PROBLEM STATEMENT

Considering that a LBS wants to estimate the future location of a user, it needs to create a predictive model of the user. In order to reach this goal, the LBS will constantly collect locations of the user to update her model as shown in Figure 1 (a). However, the location privacy of user is entirely compromised because all her raw locations are regularly shared with the LBS. This means that all sensitive information related to the user is given to a third-party entity. For example, the LBS can discover the following sensitive information related to the user from her raw locations: her home and work places but also her likes and dislikes about religion and/or politics.

The first solution is to delegate the creation of the predictive model to a service at the operating system level that we consider as trusted, as shown in Figure 1 (b). In this context, the only service that has access to the raw locations of the user coming from the positioning system is the dedicated service. The latter provides predicted locations to the LBS that needs them to operate properly. Although the location privacy of the user is increased in this context, there is still a location privacy issue about the predicted locations shared with the LBS. Because of all the predicted locations gathered by the LBS, the latter can always infer precise location habits of the user, especially when it is requesting the trusted service for the same future time every day for instance.

Consequently, the challenge is to protect as much as possible the location privacy of the user in the context of the sharing of her predicted locations with a LBS. Although there exist various LPPMs in the literature, they do not necessarily meet the utility requirement of a LBS. This means that they can easily compromise the proper functioning of the LBS until reaching the point it becomes unusable for the user. For example, the location information provided by the LBS can be inaccurate or simply erroneous because the precision of the prediction has been made too low by the LPPM. As a result, the user might stop using the LBS. As discussed in the introduction, our approach consists in building a system, including a location predictive model as well as an adapted LPPM, that takes into account the utility requirement of the LBS and the utility/privacy tradeoff expressed by the user as indicated in Figure 1 (c) or Figure 2.

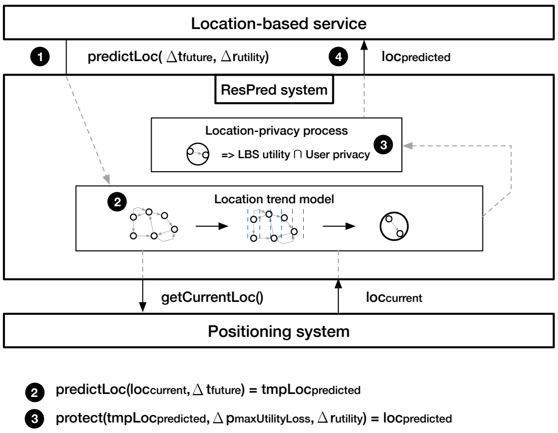

Location-based service
1
predictLoc(Δtfuture, Δrutility)
4
locpredicted
ResPred system
Location-privacy process
=> LBS utility ∩ User privacy
3
2
Location trend model
getCurrentLoc()
loccurrent
Positioning system
2 predictLoc(loccurrent, Δ tfuture) = tmpLocpredicted
3 protect(tmpLocpredicted, Δ pmaxUtilityLoss, Δ rutility) = locpredicted

Figure 2: ResPred system overview.

4 SYSTEM OVERVIEW

As described in Figure 2, *ResPred* contains two components. The first component focuses on location prediction, while the second component relates to location privacy. Consequently, the first component is responsible for the prediction of the future location of the user and includes her predictive location model, called *location trend model*. The second component aims at protecting the predicted location computed by the first component and uses a LPPM called *utility privacy tradeoff LPPM*.

A request of a LBS consists in asking where a user will be in the future. As described in Equation 1, the LBS requests the future location by specifying the time duration expressed by Δt_{future} in seconds from the current time, e.g., 7200 seconds (2 hours) from now. The LBS also indicates its required utility $\Delta r_{utility}$ that allows it to operate properly. For instance, if a LBS must call a taxi for a user in advance, the LBS will indicate an utility of a short distance in meters, such as 500 meters. A long distance could compromise the use of the taxi service itself and the related LBS because it could display inaccurate information to the user. The returned value is a location expressed by a pair $loc_{predicted} = (\phi, \lambda)$.

$$predictLoc(\Delta t_{future}, \Delta r_{utility}) = loc_{predicted} \qquad (1)$$

To summarize, *ResPred* will answer the following question: *Where will be the user in Δt_{future} second(s) from now?*

4.1 Location Prediction Component

The location prediction component contains a predictive model that represents the location trends of a user organized per time slice. As mentioned in Section 2,

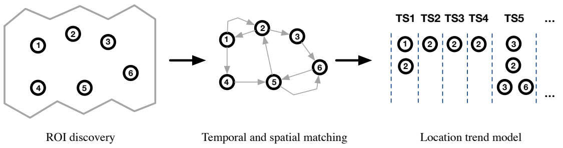


Figure 3: From ROIs to location trend model.

time is discretized into time slices during a given period of time, such as 504 time slices during one week (i.e., the duration of one time slice is 20 minutes). A location trend model is an array in which each cell contains all the possible ROIs or successive ROIs visited during a specific time slice. Figure 3 describes the creation process of the location trend model. Firstly, the *ROI discovery* process enables to discover all the ROIs of a user by analyzing the raw locations of the user. Secondly, all raw locations are marked with a specific ROI and a specific time slice as specified in the *Temporal and spatial matching* step. This step helps to pre-process the locations for the creation of the location trend model. Finally, we discover the structure of the location trend model in which we collect all the ROIs or successive ROIs visited during each time slice. Since the location trend model is a statistical model, each visited ROI or successive visited ROIs stored for a given time slice have a visit counter. This enables to highlight the location habits of the user per time slice, i.e., the ROIs or successive ROIs that are the most visited by the user during a time slice. In addition, this allows the component to find the predicted locations to answer the LBS requests.

As depicted in Figure 2, the location trend model will have to solve the following request expressed in Equation 2 and return a temporary predicted location $tmpLoc_{predicted}$. The latter is not the final predicted location sent to the LBS at the end of the process because $tmpLoc_{predicted}$ must be protected by the LPPM of the location privacy component.

$$predictLoc(loc_{current}, \Delta t_{future}) = tmpLoc_{predicted} \tag{2}$$

In order to find the $tmpLoc_{predicted}$, the location prediction component starts by searching the target time slice corresponding to the time slice that includes the future time computed by adding the Δt_{future} duration to the current timestamp, i.e., $loc_{current}.t$. After having found this target time slice, the location trend model is analyzed to find the location trends corresponding to the target time slice expressed as ROI(s). The $tmpLoc_{predicted}$ is a triplet such as $tmpLoc_{predicted} = (\phi, \lambda, \Delta r)$. Item Δr is a radius that is the accuracy of the temporary predicted location. There are two cases now to compute the items of the $tmpLoc_{predicted}$. Firstly, if the analysis highlights that the most likely visited location in the target time slice corresponds to one ROI, the temporary predicted location has the same latitude, longitude and radius as those of the ROI. Secondly, if the analysis shows that the most likely visited locations are two or several successive ROIs, the component merges all the ROIs into one single ROI and computes a new latitude, a new longitude and a new radius, which correspond to the items of the $tmpLoc_{predicted}$. In addition, it is important to note three specific location prediction scenarios that can occur during the prediction process. The best scenario is that the component finds the most likely ROI or successive ROIs to compute the $tmpLoc_{predicted}$ by exploring the location trends of the target time slice. Secondly, it can happen that all ROIs or successive ROIs have the same visit counter value. In this context, the last visited ROI or successive ROIs are used to compute the $tmpLoc_{predicted}$. Finally, it is also possible that there is no ROI or successive ROIs recorded for the target time slice. For this unique and specific problem, the component explores previous time slices until finding a visited ROI or successive ROIs to compute the $tmpLoc_{predicted}$.

4.2 Location Privacy Component

The goal of the location privacy component is to protect as much as possible the temporary predicted location found by the location prediction component. The LPPM that will be applied on the $tmpLoc_{predicted}$ depends on two aspects: the LBS utility $\Delta r_{utility}$ given by the LBS and the user privacy preference given by the user expressed as a maximum utility loss percentage $\Delta p_{maxUtilityLoss}$. This means that the LBS can provide useful and relevant information in a radius, which is the LBS utility in meters, around a reference location. Beyond this distance, there is no guarantee that the LBS is able to operate properly or to provide a reliable information to the user. For example, if the LBS is an application of a taxi company and asks a predicted location, at the end of day when the user usually requests the LBS for a taxi, in order to anticipate the user's request, the LBS will indicate a close utility in meters in order to not be far from the user in a future time. The maximum utility loss is expressed as a percentage that clearly indicates the maximum utility that the user is willing to sacrifice in order to protect her location privacy. Consequently, its value is a percentage ranged between 0 included and 1 not included. 0 is included and means that the user simply does not want to lose any LBS utility. 1 is not included because this would mean that the LBS cannot work properly if this value is reached. Equation 3 describes the request handled by the component includ-

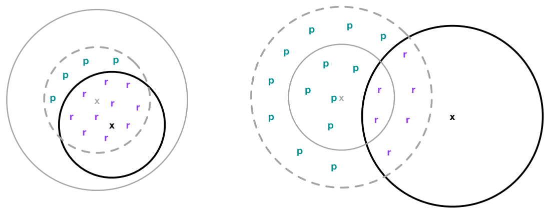


Figure 4: Computing new coordinates when the radius of the reference zone is adjusted, i.e., greater or smaller than the radius of $tmpLoc_{predicted}$.

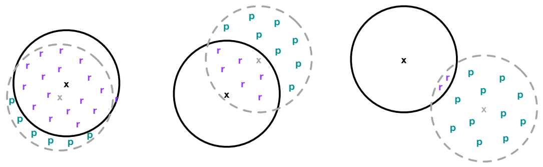


Figure 5: Three possible random generations of new coordinates (position x in gray) according to a high maximum utility loss percentage.

ing the LBS utility $\Delta r_{utility}$ and the maximum utility loss percentage $\Delta p_{maxUtilityLoss}$.

$$protect(tmpLoc_{predicted}, \Delta p_{maxUtilityLoss}, \Delta r_{utility}) = loc_{predicted} \quad (3)$$

The location privacy preserving mechanism works in the following manner. The component firstly creates a reference zone $zone_{ref}$ that has a latitude and a longitude corresponding to those of the $tmpLoc_{predicted}$ and a radius equals to the LBS utility $\Delta r_{utility}$.

The goal of the component is now to change the latitude and the longitude of the $tmpLoc_{predicted}$ by computing new coordinates. The component will create a new zone, called $zone_{new}$, which is a zone having the new generated latitude and longitude as coordinates and a radius equals to the LBS utility $\Delta r_{utility}$. In order to compute these new coordinates, the component firstly generates a random angle that indicates the direction of the new coordinates. Then, a latitude and a longitude are generated randomly in the direction of the angle between 0 and a threshold value corresponding to the case where there cannot have any intersection between $zone_{ref}$ and $zone_{new}$, i.e., $2 \times \Delta r_{utility}$. Now the component must carefully check if the protected percentage of the $zone_{ref}$ is not greater than the maximum utility loss percentage indicated by the user, i.e., $p_{maxUtilityLoss}$. In order to check this condition, the component computes the area of the intersection between the reference zone $zone_{ref}$ and the new zone $zone_{new}$. The area of this intersection is divided by the area of the $zone_{ref}$ in order to obtain a released percentage $p_{released}$, which is shared with the LBS. Finally, the component computes the protected percentage that is equal to: $p_{protected} = 1 - p_{released}$. The new coordinates are validated only if $p_{protected}$ is lower or equal to the maximum utility loss percentage given by the user. If it is not the case, new coordinates are generated until meeting this condition. When this condition is met, $loc_{predicted}$ is created with a latitude and a longitude corresponding to the new coordinates and is sent to the LBS. Therefore, there is a clear link between the utility that the user is willing to lose and her location privacy because the greater the $p_{maxUtilityLoss}$, the better the user protects her location privacy. Equation 4 summarizes the checking of this condition. The function *area* enables to compute the area of the elements passed as parameters.

$$1 - \frac{area(zone_{ref} \cap zone_{new})}{area(zone_{ref})} <= \Delta p_{maxUtilityLoss} \quad (4)$$

This means that the location privacy component tries to find an appropriate tradeoff between LBS utility and the location privacy preference chosen by the user. In order to illustrate the process, Figure 4 depicts the impact of the change of the radius of the reference $zone_{ref}$ in the case where the $tmpLoc_{predicted}$ has a radius greater than $\Delta r_{utility}$ and in the case where $tmpLoc_{predicted}$ has a radius smaller than $\Delta r_{utility}$. The $tmpLoc_{predicted}$ is the gray circle with the dotted lines, the $zone_{ref}$ is the gray circle and the dark circle is the $zone_{new}$. The center of the $zone_{new}$ corresponds to the location that is sent to the LBS by *ResPred*. The letters r indicate the zone that is released to the LBS, while the letters p describe the zone that is protected. In addition, Figure 5 depicts three possible random generations of new coordinates according to a maximum utility loss percentage that is really high. The resulting value of the process of this component is the predicted location $loc_{predicted}$, which is also returned to the LBS as described in Equation 3 and in Figure 2.

5 EVALUATION

The main goal of the evaluation is to assess our system from a utility and a location privacy perspective. In order to reach this goal, we ran several experiments taking into account different LBS scenarios and different LPPMs including our mechanism and existing ones. In addition, we also compute the location prediction accuracy of the location trend model.

5.1 Dataset

We chose real user locations of two datasets: *PrivaMov* dataset described in (Ben Mokhtar et al., 2017) and a very detailed dataset of one user. From these two datasets, we extracted locations of users that were captured via different positioning system such as GPS, radio cells as well as WiFi. We performed an analysis in order to select the best users of *PrivaMov* for our evaluation. This selection was based on the quality of the user datasets. This quality was assessed by computing the percentage of hours during one day having at least one location, called *daily percentage* later. In order to properly fill the location trend model, we need very rich user datasets without important gaps in terms of days. More specifically, a user is selected if the average of all her daily percentages is greater or equal to 0.4, if her dataset duration in terms of days was greater or equal to 30 days and lower than 250 days and if all weekdays (from Monday to Sunday) have at least one daily percentage. Seven users only of the *PrivaMov* dataset met all these conditions. Consequently, we evaluated eight users in total, i.e., seven users from *PrivaMov* and one user from the private dataset. The average of the duration of all evaluated user datasets is 115 days and the average of the number of locations of all evaluated user datasets is 7’580’391.

5.2 LBS Scenarios

We decided to define two LBS scenarios for the evaluation. The first scenario is a public transportation LBS that provides next departure information of bus, metro and train in advance. The information is displayed on the mobile of the user just before the usual checking of the public transportation departures by the user. The second scenario is a taxi LBS that calls a taxi in advance for the user by following the usual use of the service by the user. We assume that the LBS knows the usage habits of the service by the user but it does not obviously know the location of the user in the future. That is why these two LBS must use our *ResPred* system to obtain it. As depicted in Figure 2, an LBS can request a user’s location in the future, e.g., 2 hours from now. For each scenario, we chose an array of target time slices for which the predicted locations must be computed. The parameters of these two LBS scenarios are defined in Section 5.4.

5.3 Existing Location Privacy Preserving Mechanisms

In order to properly assess the LPPM of our system, we selected two existing LPPMs from the literature. We compare them with our mechanism from the utility/privacy perspective whose metrics are detailed in Section 5.6. We chose the spatial rounding presented in (Krumm, 2007; Agrawal and Srikant, 2000) as well as the Gaussian perturbation described in (Armstrong et al., 1999). The spatial rounding works with a grid that discretizes the space in which the user is moving. The mechanism transforms raw coordinates of a location into new coordinates corresponding to the nearest vertex of the square or rectangle that is a grid’s cell in which the raw location is. The spatial Gaussian perturbation is a mechanism that adds spatial noise to the latitude and the longitude of a raw location according to a certain mean and a standard deviation. All these parameters are presented in next section.

5.4 Experimental Settings

The experimental settings of the utility/privacy evaluation as well as location prediction accuracy evaluation are detailed in this section.

5.4.1 ROI Discovery

In order to discover the ROIs of a user, we use a specific part of a discovery process of Zones of Interest (ZOIs) described in (Kulkarni et al., 2016). The Δd_{max} is equal to 60 meters and Δt_{min} has a value of 10 minutes. We follow the creation process of clusters, which are called ROIs in this paper, without creating any cluster groups or ZOIs similarly to (Kulkarni et al., 2016). After discovered all clusters, we merge them if an intersection occurs between two clusters and we repeat this merging process until reaching a stable cluster set in which there is no more intersection. We do not filter out the ROIs that are not frequently and/or not recently visited because we want to keep a high number of ROIs describing the mobility of the user, in order to properly fill the location trend model.

5.4.2 Location Trend Model

The location trend model is created with time slices having a duration of 20 minutes during a period of one week, resulting in 504 time slices for one week. We chose this time slice duration by exploring the entropy level of each time slice cell of the location trend model and finding that it was the best time slice duration for the location prediction goal.

Table 1: List of parameters used for each LBS scenario.

Parameter/LBS scenario	Public transportation LBS scenario	Taxi LBS scenario
LBS utility distance	1000 meters	500 meters
Number of target time slices	10	4
Frequency of requests	100	100
Random repartition of predicted locations	0 and 1	0 and 1

5.4.3 LBS Scenarios

A scenario corresponds to a specific type of LBS as described in Table 1: the public transportation LBS and the taxi LBS. Firstly, each LBS has its own utility distance and a specific number of target time slices. For example, the target time slices of the public transportation LBS are in the morning, i.e., from 7:00 to 7:20 am, and at the end of the afternoon, i.e., from 5:00 to 5:20 pm, every working day. Regarding the taxi LBS, the target time slice are in the evening Thursday from 10:00 to 10:20 pm, Friday from 11:00 to 11:20 pm, and Saturday from 4:00 to 4:20 pm and from 11:00 to 11:20 pm. We distribute the total number of location prediction requests, called *frequency* in Table 1, which are 100 in total, per target time slice for each LBS scenario. The distribution of the total number of location prediction requests can be equal for all the target time slices, i.e., 10 (100 divided by 10) for each target time slice. Or the process can also randomly distribute the 100 predicted location requests per target time slice meaning that some time slices can have more predicted locations than others. These scenarios are the same for all evaluated users.

5.4.4 Location Privacy Preserving Mechanisms

As mentioned previously, we chose two LPPMs: the grid-based rounding and the Gaussian perturbation in addition to our proposed LPPM. For the *utility privacy tradeoff LPPM* included in *ResPred*, we selected four values for $\Delta p_{maxUtilityLoss}$: 0.2, 0.4, 0.6 and 0.8. Regarding the grid-based rounding, we decided to have a difference of 0.005, 0.05 and 0.5 between two successive latitudes or longitudes to create each cell of the grid. The values of the grid-based parameters are ranged from approximately 380 to 38000 meters. Finally, we chose 4 standard deviations that are 0.0005, 0.005, 0.05 and 0.5 for the Gaussian perturbation, the mean being the latitude or the longitude of the raw location. The values of the Gaussian perturbation parameters are ranged from approximately 55 to 66500 meters.

5.5 Location Prediction Accuracy

We decide to evaluate the location prediction accuracy of the location trend model by performing the following steps. Firstly, the dataset of each user must be divided into two datasets according to the total number of locations: a training set of 60% and a test set of 40%. We discover the ROIs and we create the location trend model of a user with her training dataset. Secondly, the evaluation process is the following: we start by selecting 200 unique locations in the test set. For each selected location of the test set, we convert its timestamp into a target time slice. Then, we find the most likely visited ROI or successive ROIs of the target time slice by exploring the location trend model, which is the same process explained in Section 4.1. We consider that the prediction is correct for this location only if the latter is contained in the ROI or the merged ROI computed from the successive found ROIs. If there is no ROI for the target time slice, we simply do not take the prediction into account. At the end, we compute a ratio that is the number of correct predictions out of the number of predictions that returned a value after having explored the model.

5.6 Utility/Privacy Metrics

The metrics presented in this section enables to evaluate the utility as well as the location privacy of all predicted locations of a user shared with a LBS and, consequently, to highlight the LPPM that gives the best utility/privacy tradeoff.

5.6.1 Utility Metric

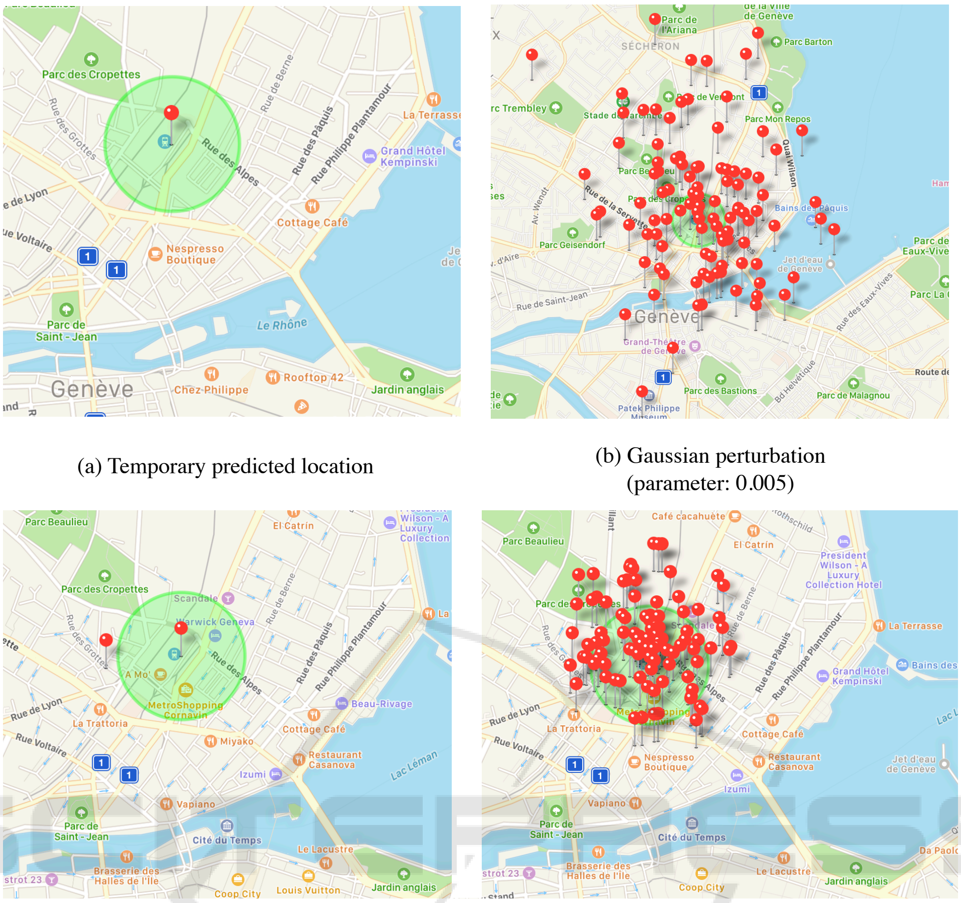
The utility metric allows us to evaluate if a predicted location sent to the LBS meets the utility requirement of the LBS given at the beginning of the process by the LBS itself. We define a reference zone $zone_{ref}$ that has a center corresponding to the center of the $tmpLoc_{predicted}$ and a radius that has the value of $\Delta r_{utility}$. We also create a zone to check $zone_{toCheck}$ having a center that is the coordinates of the predicted zone $loc_{predicted}$ and a radius equals to the value of $\Delta r_{utility}$. The utility is validated if there is an intersection between the $zone_{ref}$ and the zone to check $zone_{toCheck}$. Equation 5 describes the two possible results of the utility metric.

$$res_{utility} = \begin{cases} 0, & \text{if } zone_{ref} \cap zone_{toCheck} = \emptyset \\ 1, & \text{if } zone_{ref} \cap zone_{toCheck} > 0 \end{cases} \quad (5)$$

Then we compute the utility average of a target time slice by dividing the number of predicted locations that meet the utility condition by the total number of predicted locations sent to the LBS for this target time slice. Finally, we calculate the average of the utility results obtained for all the target time slices in order to obtain the utility result of the scenario.

5.6.2 Location Privacy Metric

The location privacy metric corresponds to a metric that evaluates the degree of confusion of an adversary, the LBS in our case, during a location attack on the predicted locations received from *ResPred*. The metric is based on the Shannon entropy that can compute



(a) Temporary predicted location

(b) Gaussian perturbation (parameter: 0.005)

(c) Rounding mechanism (rounded to 2 decimals)

(d) Utility privacy tradeoff LPPM (parameter: 0.9)

Figure 6: Visual description of the impact of the different LPPMs on a temporary predicted location.

a level of uncertainty as described in (Shokri et al., 2011). As mentioned in Section 2.4, the location attack performed by an adversary consists in trying to discover one location amongst all of the predicted locations sent by *ResPred* for a specific target time slice considering that the adversary knows how the time is discretized in our location trend model. The goal of a LPPM is to confuse the adversary in order to reduce its probability of finding one single location for a target time slice. In order to compute the location privacy, we will create a grid that discretizes the space and compute the density proportion $p_{density}$ of each visited cell of this grid. The density proportion $p_{density}$ is the number of predicted locations out of the total number of predicted locations visited per visited cell of the grid during the target time slice. Each cell of the grid is a rectangle of approximately 100 meters per 180 on an average, i.e., a difference of 0.001 between two successive latitudes or longitudes. Equation 6 describes the computation of the location privacy for a specific target time slice in which i is the index of the i^{th} visited cell by the user, n is the total number of visited cells by the user during the target time slice. A low entropy result means a low confusion of the adversary, while a high entropy result means a high uncertainty.

$$res_{locationPrivacy} = -\sum_{i=1}^{n} p_{density_i} \log_2 p_{density_i} \qquad (6)$$

Finally, we compute an average result for each scenario in the same way as for the utility metric (described at the end of the previous section).

5.7 Results

The average of the location prediction accuracy of the location trend model for all evaluated users is equal to 60%. In addition, we obtain a minimum and a maximum location prediction accuracy of 16% and 90% respectively.

Table 2: Utility / location privacy results.

LPPM/Result	Utility result	Location privacy result
Utility privacy tradeoff LPPM	**1.0**	**2.81**
Grid-based rounding	0.62	0
Gaussian perturbation	0.50	2.78

Regarding the utility/privacy tradeoff evaluation, we firstly compute the average of the utility results of all LBS scenarios per user and, secondly, we calculate average utility results of all users. We do exactly the same for the location privacy results. The results are summarized in Table 2. We can clearly see that our LPPM, i.e, the utility/privacy tradeoff LPPM, has the best utility/privacy tradeoff because the utility result and the location privacy result reach the highest values. This means that our LPPM meets the LBS utility requirements and is also able to protect the location privacy of the user according to her privacy preference. Although the Gaussian perturbation has also a high location privacy result, a reasonable utility result is not reached. The location privacy result of the grid-based rounding is equal to 0 indicating that the adversary has no confusion because the modified locations, i.e., predicted locations, are always the same for a target time slice. The Gaussian perturbation has the advantage of blurring the location via a single parameter expressing a distance, while the grid-based mechanism requires the creation of a grid that can take a substantial time and its exploration before being able to blur a location. Although our mechanism must check a location privacy condition, it computes the new coordinates within a reasonable time.

Finally, we can see the blurring impact of the different LPPMs on a temporary predicted location in Figure 6. In Figure 6 (a), we can see the center as well as the radius of a temporary predicted location, both depicted with a marker and a circle. In Figure 6 (b) and (d), 100 new locations, depicted with new markers, are created according to the corresponding LPPM. Regarding the rounding, the coordinates have only been rounded to two decimals in the figure but in the context of the evaluation with a spatial grid, we would have obtained 100 times the same location because the structure of the grid is fixed and the nearest location is always the same for a single location to blur.

6 RELATED WORK

The related work below tackles the two main subjects of the paper that are the following: the description of existing LPPMs as well as the different predictive models presented in the literature that are used to compute future user locations.

6.1 Location Privacy Preserving Mechanisms

In a location prediction context, we consider that we need to protect the predicted location that is sent to a LBS as mentioned in Section 3. To reach this goal, there exist various mechanisms to protect the predicted location, such as applying a spatial perturbation (Agrawal and Srikant, 2000; Armstrong et al., 1999; Gambs et al., 2011), using a spatial cloaking mechanism (Gruteser and Grunwald, 2003), sending dummy locations (Kido et al., 2005) or using a rounding mechanism (Agrawal and Srikant, 2000; Krumm, 2007).

Applying a spatial perturbation enables to spatially modify a location as mentioned by several authors in (Armstrong et al., 1999; Gambs et al., 2011). As described in these papers, we can add spatial noise to the coordinates of a location. However, the more noise is added to the location sent to the LBS increases, the more the LBS utility decreases in our context because a LBS may provide information that is not related to the raw predicted location, depending on the level of protection. In the case of the spatial cloaking presented by Gruteser and Grunwald in (Gruteser and Grunwald, 2003), the predicted location should only be sent if the user is considered as *k-anonymous*, meaning that the user cannot be distinguishable from at least $k-1$ other users. This technique is unfortunately not realistic in our context and not easy to implement especially in the case where the mobility models of users are not centralized or shared in a common server. As detailed in (Kido et al., 2005), sending dummy locations is interesting in order to add noise if and only if multiple predicted locations can be sent to a LBS. However in our system, it is impossible to use this LPPM because only one predicted location must be sent to a LBS as an answer to a predictive request supported by *ResPred*. Utilizing a rounding mechanism, as described in (Agrawal and Srikant, 2000; Krumm, 2007), can be considered because the predicted location is changed into a new location corresponding to a nearest reference point. If we consider that space is discretized and described with multiple reference points (the vertices of each cell of a grid for instance), the mechanism consists in modifying a location into a new location corresponding to the nearest reference vertex of the cell in which the location is as indicated in the papers cited previously. Cryptography techniques could be also used to protect locations sent to third parties as mentioned in (Hendawi and Mokbel, 2012) but our work is not focused on this kind of privacy/security strategies. To summarize and according to the best of our knowledge, there is no LPPM that can find an appropriate tradeoff between the utility and the privacy in a location prediction context. For our utility/privacy evaluation, we chose the closest LPPMs to our work, that are the rounding and the spatial perturbation as detailed in the previous section.

6.2 Location Prediction Requests and Models

As detailed in the complete survey in (Hendawi and Mokbel, 2012), various techniques exist to predict future locations of users. In the literature, there exist different location predictive models for different types of location prediction requests, such as predicting a future location based on a time duration (Jeung et al., 2008; Sadilek and Krumm, 2012), predicting the next location that will probably be reached by a user (Gambs et al., 2012; Gidófalvi and Dong, 2012; Ying et al., 2011), etc. Some location prediction-based papers focus on other location-based predictive requests, such as the prediction of the staying time in a particular ROI or when the user will reach or leave a ROI (Gidófalvi and Dong, 2012), the prediction of the number of users reaching a specific zone (Chapuis et al., 2016) and much more. Other remaining works are focused on range queries that enable to identify if one or multiple user(s) will be in a specific area during a specific time window. In (Xu et al., 2016), the authors describe a way to prune an order-*k* Markov chain model in order to efficiently compute long-term predictive range queries.

The main focus of our paper, in terms of prediction, is to comnpute a future location of a user based on a time duration from the current time. In the literature, it is shown that some predictive models can work better for near location predictions and others are more suited for distant location predictions. In (Jeung et al., 2008), the authors present a hybrid prediction model for moving objects. For near location predictions, their model uses motion functions, while for distant location predictions, their model computes the predicted location based on trajectory patterns. The structure in which they store the trajectory patterns of a user is a trajectory pattern tree. However, they do not evaluate their model with real mobility traces. Their predictive model is close to our location trend model because they use the notion of patterns based on spatial clusters to fill their model. Nevertheless, the structure of their final model is clearly not the same as ours because they create a trajectory pattern tree. Sadilek and Krumm propose a method to predict long-term human mobility in (Sadilek and Krumm, 2012) up to several days in the future. Their method, which can highlight strong pattern of users, uses a projected eigendays model that is carefully created by analyzing the periodicity of the mobility of a user as well as other mobility features. This work highlights that it is crucial to extract strong patterns for long-term predictions. The location trend model we propose in the *ResPred* system is close to the model presented by Sadilek and Krumm. However, our model is different in that it is based on ROIs and not on raw locations and takes less features into account.

7 CONCLUSION

In this paper, we presented a system called *ResPred* that enables to compute predicted locations of a user for LBS. This system contains two components. The first component focuses on location prediction by including a predictive model based on location trends expressed as ROI(s). The second component aims at protecting the location privacy of the user by finding an appropriate tradeoff between a utility specified by the LBS and a location privacy preference indicated by the user that is expressed as a maximum utility loss percentage. The results clearly show that our LPPM provides the best utility/location privacy tradeoff compared to two other existing LPPMs. In addition, the location trend model is promising if we look at the location prediction accuracy results, especially in the context of location prediction according to a certain time duration in the future. Future work will consist in extending the evaluation to more users by finding a dataset having rich user datasets, which is a real need for the research community. We will also design other inference attacks in order to evaluate the location privacy and maybe compare the computing cost of the different LPPMs. And finally, we will compare the location trend model to other existing close models for similar requests regarding short, mid and long-term location predictions.

REFERENCES

Agrawal, R. and Srikant, R. (2000). Privacy-preserving data mining. In *ACM Sigmod Record*, volume 29, pages 439–450. ACM.

Armstrong, M. P., Rushton, G., and Zimmerman, D. L. (1999). Geographically masking health data to preserve confidentiality. *Statistics in Medicine*, vol. 18:497–525.

Ben Mokhtar, S., Boutet, A., Bouzouina, L., Bonnel, P., Brette, O., Brunie, L., Cunche, M., D 'alu, S., Primault, V., Raveneau, P., Rivano, H., and Stanica, R. (2017). PRIVA'MOV: Analysing Human Mobility

Through Multi-Sensor Datasets. In *NetMob 2017*, Milan, Italy.

Chapuis, B., Moro, A., Kulkarni, V., and Garbinato, B. (2016). Capturing complex behaviour for predicting distant future trajectories. In *Proceedings of the 5th ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems*, pages 64–73. ACM.

Gambs, S., Killijian, M.-O., and Núñez del Prado Cortez, M. (2011). Show me how you move and i will tell you who you are. *Trans. Data Privacy*, 4(2):103–126.

Gambs, S., Killijian, M.-O., and Nuñez Del Prado Cortez, M. (2012). Next place prediction using mobility Markov chains. In *MPM - EuroSys 2012 Workshop on Measurement, Privacy, and Mobility - 2012*, Bern, Switzerland.

Gidófalvi, G. and Dong, F. (2012). When and where next: individual mobility prediction. In *Proceedings of the First ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems*, pages 57–64. ACM.

Gruteser, M. and Grunwald, D. (2003). Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the 1st international conference on Mobile systems, applications and services*, pages 31–42. ACM.

Hendawi, A. M. and Mokbel, M. F. (2012). Predictive spatio-temporal queries: A comprehensive survey and future directions. In *Proceedings of the First ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems*, MobiGIS '12, pages 97–104, New York, NY, USA. ACM.

Jeung, H., Liu, Q., Shen, H. T., and Zhou, X. (2008). A hybrid prediction model for moving objects. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 70–79. Ieee.

Kido, H., Yanagisawa, Y., and Satoh, T. (2005). An anonymous communication technique using dummies for location-based services. In *Proceedings of the International Conference on Pervasive Services 2005, ICPS '05, Santorini, Greece, July 11-14, 2005*, pages 88–97.

Krumm, J. (2007). Inference attacks on location tracks. In *Proceedings of the 5th International Conference on Pervasive Computing*, PERVASIVE'07, pages 127–143, Berlin, Heidelberg. Springer-Verlag.

Kulkarni, V., Moro, A., and Garbinato, B. (2016). A mobility prediction system leveraging realtime location data streams: poster. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pages 430–432. ACM.

Sadilek, A. and Krumm, J. (2012). Far out: Predicting long-term human mobility. In *AAAI*.

Shokri, R., Theodorakopoulos, G., Le Boudec, J.-Y., and Hubaux, J.-P. (2011). Quantifying location privacy. In *Proceedings of the 2011 IEEE Symposium on Security and Privacy*, SP '11, pages 247–262, Washington, DC, USA. IEEE Computer Society.

Xu, X., Xiong, L., Sunderam, V., and Xiao, Y. (2016). A markov chain based pruning method for predictive range queries. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '16, pages 16:1–16:10, New York, NY, USA. ACM.

Ying, J. J.-C., Lee, W.-C., Weng, T.-C., and Tseng, V. S. (2011). Semantic trajectory mining for location prediction. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 34–43. ACM.

Zang, H. and Bolot, J. (2011). Anonymization of location data does not work: A large-scale measurement study. In *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking*, MobiCom '11, pages 145–156, New York, NY, USA. ACM.