Location privacy attacks based on distance and density information

Sergio Mascetti
Department of Computer Science
University of Milan
sergio.mascetti@unimi.it

Letizia Bertolaja
Department of Computer Science
University of Milan
letizia.bertolaja@unimi.it

Claudio Bettini
Department of Computer Science
University of Milan
claudio.bettini@unimi.it

ABSTRACT

Proximity services alert users about the presence of other users or moving objects based on their distance. Distance preserving transformations are among the techniques that may be used to avoid revealing the actual position of users while still effectively providing these services. Some of the proposed transformations have been shown to actually guarantee location privacy with the assumption that users are uniformly distributed in the considered geographical region, which is unrealistic assumption when the region extends to a county, a state or a country.
In this paper we describe a location privacy attack that, only using partial information about the distances between users and public knowledge on the average density of population, can discover the approximate position of users on a map, independently on the fake or hidden position assigned to them by a privacy preserving algorithm. We implement this attack with an algorithm and we experimentally evaluate it showing that it is practically feasible and that partial distance information like the one exchanged in common friend-finder services can be sufficient to violate users' privacy.

Categories and Subject Descriptors

K.4.1 [**Computers and Society**]: Public Policy Issues—*Privacy*; K.6.5 [**Management of Computing and Information Systems**]: Security and Protection

General Terms

Design, Experimentation, Security

Keywords

Location Privacy, Proximity-Based Services, Distance preserving transformations

1. INTRODUCTION

The problem of privacy preservation in location based services has been extensively studied in the literature with particular focus on services aimed at identifying points of interest (e.g., pubs, hotels, gas stations) close to the user [1, 3]. A class of services, called *friend-finders*, aims at notifying a user when a "friend" happens to be spatially closer than a certain threshold distance. These services can be efficiently implemented by a server receiving position information from each user and sending an alert when the distance threshold condition is verified. Since location information can be considered private information, a user may not want to disclose his exact position to the service provider. Indeed, we consider the service provider a semi-honest entity: it follows the given protocol, but it may attempt to infer the actual locations of the users. An intuitive way to preserve privacy is to use fake locations that are computed in such a way that they preserve (possibly with an approximation) the relative distances between users. With this solution the proximity service can be still provisioned, and the precise coordinates of users are not disclosed to the service provider. Distance preserving transformations are safe (i.e., they don't reveal the users' actual positions) if no background knowledge is assumed, including the distribution of the users. This assumption is implicitly or explicitly made in several papers as briefly described in the following, however it is unclear in the literature whether the only knowledge of users' distribution could lead to privacy violations in practical cases.

Among the protocols for friend-finder services, the *Louis* protocol [11] allows the server to obtain proximity information between users and it is assumed that this information may not lead to a location privacy breach. This assumption should be probably reconsidered in light of our results. A similar strategy is adopted by the *FriendLocator* protocol [9] that introduces a significant optimization in the proximity updates but that still reveals to the server some approximate information about the distance of users. The *Longitude* protocol [6] suffers from the same privacy problem, and indeed its safety relies on the assumption that the service provider has no a priori knowledge on the distribution of users, i.e., a uniform distribution is assumed. A more recent solution [8], called *VicinityLocator*, includes several improvements over the *FriendLocator* protocol among which a finer control of users' privacy preferences similar to that of *Longitude*. However, *VicinityLocator* still reveals to the server approximate distance information.

The possible unsafety of distance preserving transformations has been shown in previous work. In the context of privacy preserving data mining, it has been shown that distance preserving transformations can be subject to privacy attacks in

ACM SIGSPATIAL GIS '12, November 6-9, 2012. Redondo Beach, CA, USA
Copyright (c) 2012 ACM ISBN 978-1-4503-1691-0/12/11...$15.00

presence of prior knowledge of the adversary [5] in terms of a subset of input-output pairs and samples from original or similar dataset. The concept of distance in that work is more general than the geographical distance on a map and applied to general attribute values when Euclidean distance can be computed. In our attack the only background knowledge is the publicly available density distribution. The unsafety of distance preserving transformations in presence of certain kinds of background knowledge is also discussed in the context of secure kNN queries [10]. The authors consider as safe the case in which the adversary observes only the encrypted DB and the distances (called "level 1" attack). Our study shows that even this attack model can lead to privacy breaches when the distribution of the original data is known, as it is the case for population distribution.

In the context of private spatial join computation, a distance based attack in presence of background knowledge about spatial distribution is described in [4], showing that the position of isolated points (outliers) may be discovered. A defense is proposed based on a spatial transformation that eliminates distances longer than a certain threshold in the transformed space. In principle, friend-finder services may be effective even if limited to short distances (2-3 km); however, our study shows (see Section 4) that even ignoring any distance larger than that threshold, the attack we propose can still pose privacy threats. In addition, our study does not want to rule out services in which long distances may have to be considered, and identifies under which conditions the adversary may actually derive users location.

To the best of our knowledge this is the first study that systematically explores the problem of identifying the position of moving objects based only on partial information on their distance and on background knowledge on the density distribution. We provide a model of the problem in terms of a privacy attack in presence of background knowledge, and we describe a clustering based algorithm implementing the attack. Our experiments consider real geographic data of France and publicly available population density data about that region, showing that the attack can easily identify the approximate position of users.

2. PROBLEM MODELING

Our scenario assumes that a service is provisioned to a set of users by exchanging data through a third party that acquires only information about the distance between some of the users. We consider this party a semi-honest adversary, i.e., following the rules of the service protocol, but at the same time using the acquired distance information as well as background knowledge to identify the geographical position of the users.

Let U be the set of users. Given a user $u \in U$, his position is a point in a discrete bi-dimensional space S. We denote with $d_p(p_1, p_2)$ the distance between two points $p_1, p_2 \in S$. Given the average density of population in each area at the precision available from public sources, we consider as *background knowledge* BK the information about how many users are in each area. More formally, the adversary will be able to compute the function $nu : G \rightarrow \mathbb{R}$ returning the number of expected users contained in g for each cell g of a grid G (depending on BK) partitioning the spatial domain S. Given $nu()$, the probability that a generic user is located in a cell g is given by the number of users in g divided by the total number of users $|U|$. During the provisioning of the service the adversary can collect data about the distances between users: we call this information the *observation knowledge* OK. Note that the distance information in OK is partial, i.e., limited to certain pairs of users.

We now show how the adversary can exploit BK and OK to infer the approximate position of the users. One approach is to compute a mapping that assigns a geographical position to each user, satisfying the distance constraints observed in OK. Then, since from BK it is possible to compute the probability of a user being in a cell of G, it is possible compute the "combined" probability of a mapping as the product of probabilities for each user. A major limitation of this approach is due to the combinatorial explosion of the number of possible mappings that is exponential in the number of users. Despite this limitation, we now describe a practical location privacy attack composed of two steps: a) users are clustered according to their distances based on OK, hence identifying sets of users close to each other, and b) clusters are mapped to geographical regions based on BK. As an example, clusters of appropriate dimensions may correspond to cities, and the adversary may be able to associate a cluster of users to the correct city as their actual approximate position. Intuitively, this attack is practically feasible when the number of clusters and regions is small, as in the case of the main cities of a country.

In this attack we model an arbitrary geographical region z as the set of cells from G. Intuitively, these regions are candidate regions where the adversary tries to locate users. After the clustering, we need to associate each cluster with the correct region, so that we can derive the geographical position of the cluster. We say that a cluster-to-region mapping m, representing an association of each cluster to a region, is *consistent* if for each pair of clusters and each pair of users in the two clusters whose distance is known, the distance between the two users is smaller (greater, respectively) than the maximum (minimum, respectively) distance between the two regions *corresponding* to the two clusters. We only consider consistent mappings.

Since an attack should indicate the most likely mapping, we want to assign a probability to each consistent cluster-to-region mapping. We define the probability that a generic user u is located in region z as:

$$P[u, z] = \sum_{g \in z} \frac{nu(g)}{|U|}$$

Since the probability that a user is in a region z is independent from the probability of other users being in z and each user has the same probability, the probability $P[c, z]$ that all users in a cluster c are in z is given by: $(P[u, z])^{|c|}$.

Given the set of clusters C and a generic user u, we define the absolute probability that a consistent cluster-to-region mapping m is correct as:

$$P[m] = \prod_{c \in C} (P[u, m(c)])^{|c|}$$

The relative probability $P_r[m]$ for mapping m can be defined as the absolute probability $P[m]$ divided by the sum of the absolute probabilities of all consistent cluster-to-region mappings. The attack selects the mapping with the highest probability.

3. TECHNIQUES

In this section we show how the attack based on user clustering can be computed in practice. We compute the function $nu(g)$ from the distribution of inhabitants of g, which is public information [2], and by assuming that the users of the service are uniformly distributed in the population. Formally: $nu(g) = (inhab(g) \cdot |U|)/totPop$, where $inhab(g)$ is the number of inhabitants of g and $totPop$ is the total population in the spatial domain S.

The computation of k clusters of users needs to be based only on the observation knowledge OK, i.e., on the relative distances between some pairs of users. We adopt the *Single Linkage* clustering technique that indeed relies only on relative distances [7]. Differently from Single Linkage, our algorithm terminates in three cases: a) when there is a single cluster, b) when no distance is known between identified clusters and c) when there are at least k clusters, each one containing at least n users (where n is a parameter of the algorithm). The first two conditions are called "failed clustering", while the third indicates a "successful clustering".

Once clusters of users have been identified, cluster-to-region mappings should be generated and the consistent ones selected. One problem arises in this process due to what we call "cluster stretching": when the clustering algorithm is executed, the large majority of the users in a cluster are located in a region z, but in some cases few of them are also located slightly outside the region (e.g., in the suburbs of a city). The "cluster stretching" problem could lead to erroneously classify a mapping as inconsistent. To face this problem, we introduce a tolerance error factor $\alpha \in (0, 1)$ in the condition to check the consistency. The intuition is that, when α is close to one, the tolerance is low. For values of α closer to 0, the consistency condition becomes easier to satisfy, even if some users are slightly outside the two regions.

The last step of the attack process consists in computing the absolute and relative probabilities of each consistent cluster-to-region mapping and selecting the one with highest probability. While, in theory, the number of consistent cluster-to-region mappings can be in the same order of all the possible cluster-to-region mappings, in practice these mappings are a small fraction (in our experiments, in which major cities are considered as regions, they are in the order of a few units, or tens, at most).

4. EXPERIMENTS

Our experiments simulate the observation knowledge (OK) that an adversary may obtain by running a friend-finder service in which privacy protection is implemented through a distance preserving transformation. The aim of the attack is to infer the actual position of users (supposed to be hidden to the service) in terms of the city in which they are located.

The overall structure of our experimental evaluation is the following: we first simulate the position of some users in a geographical area. Then, we compute the distance between some pairs of users and we use this data to perform the attack. Finally, using the original information about users' position, we evaluate the correctness of the attack. The results are computed as the average, as well as minimum and maximum, out of 100 runs.

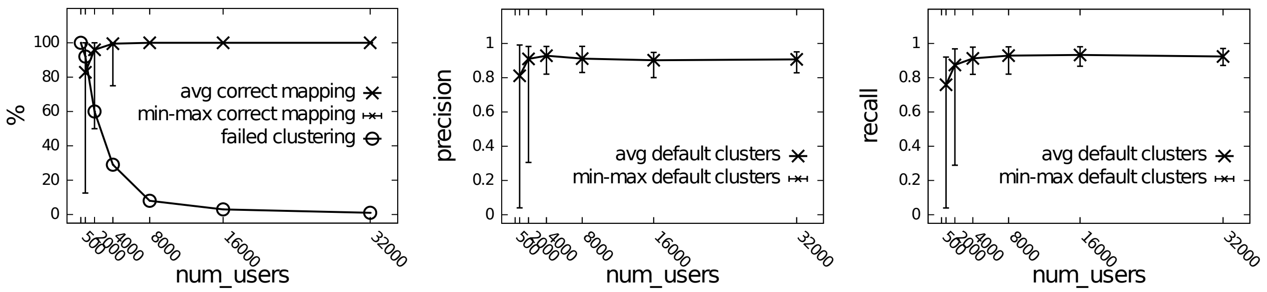
The simulation of users' position consists in randomly choosing a point in the geographical area according to a probabilistic distribution, taken from GPWv3 [2], a dataset that provides density information by dividing the world into cells and providing the number of inhabitants for each cell. Since no public information is available at a higher resolution (a cell edge is $\approx 5km$ at the equator), in the current setup we assume that, within each cell, the population density is uniform. In the experiments we consider a geographical area corresponding to France and we vary the number of simulated users "num_users" between 500 and 32000.

Another parameter is the number of pairs of users for whom their distance is defined (fixed to 80). Since our reference scenario is a friend-finder service, in order to capture the intuition that users tend to have more friends in a close-by region, 50% of the users' distances we generate are closer than 100 kilometers. Parameters k and α are fixed to 8 and 0.75 respectively, while the number of considered cities is 11. Using these values and 16000 simulated users, we can compute an attack in about 2 minutes on a 2.26GHz CPU with 4GB of main memory.

The aim of the experimental evaluation is to assess the attack effectiveness in terms of correctness of the "cluster-to-city association" and "users-to-city association" that an adversary may obtain. The "cluster-to-city association" measures the percentage of clusters that are correctly associated to the corresponding city. A cluster is considered correctly assigned to a city if at least 50% of its users are located within the assigned city's MBR. The "user-to-city association" measures the precision and recall of the association between a user and a city. The association of a user with a city is correct if the user is actually located within a city's MBR and is assigned by the attack to that city.

For what concerns the cluster-to-city association we can observe from Figure 1(a) that, for *num_users* smaller than 8000, it is relatively frequent to have a successful clustering but a wrong cluster-to-city association. Vice versa, for larger values of *num_users* the cluster-to-city association is always correct. This is an important property of our attack: when the number of observed users is sufficiently high if the clustering does not fail, then the adversary knows with high likelihood that the cluster-to-city association is correct.

The experiments to measure the user-to-city association show that, for sufficiently high values of *num_users*, the users that are assigned to a city have a high probability to be actually located within that city's MBR. Consider Figure 1(b): for values of *num_users* equal to or larger than 2000 the average precision is almost not affected by the value of *num_users* and the average precision is always above 90%. However, for values of *num_users* less than 4000, the variance is high. The last set of experimental results computes the recall in the user-to-city association, i.e., the fraction of users that are associated to a city with respect to the ones that actu-



(a) Cluster-to-city association

(b) Users-to-city precision

(c) Users-to-city recall

Figure 1: Experimental results considering France

ally should be associated. As shown in Figure 1(c), when varying the number of users, we can observe a phenomenon similar to what we observed for the precision. Indeed a small number of users only partially affects the average value of the recall. However, when the number of users is below 4000, the problem is that the variance of the recall is higher, due to the wrong cluster-to-city association.

We have some final observations and comments about the experimental results. Our attack can be considered as the first step of a more extended attack. For example, assume that the users that are associated to a city are used as pivot to trilaterate the position of other users. Second, we run a set of experiments to evaluate the effectiveness of our attack when only short distances (less than $3km$) are known to the adversary. This is the situation that would result when using a defense like the one proposed in [4]. We adapted our attack to this situation by changing the clustering algorithm so that it terminates when the distance between the two closest clusters is larger than 3km. Our results show that the attack is still effective in most of the cases: using $num\_users = 16000$, $k = 2$ and 4 cities, we obtained 100% of correct cluster-to-city associations. On the other hand this defense poses some challenges to our attack both from the computational point of view, and from its effectiveness, since it reduces the number of possible users whose position can be directly identified.

5. CONCLUSIONS

This paper describe how a practical attack can be performed to approximate locate moving objects based only on partial relative distance information and public knowledge about objects distribution. Our experiments consider a realistic scenario in which distances could be acquired by a friend-finder service guaranteeing location privacy through (any) distance preserving transformation, and assuming as the only background knowledge public data on population density. We show that a relatively low number of distances is sufficient to correctly position the users located in the major cities of a country. Future work includes refining the attack to reduce the candidate location of users, extending experiments to geographical areas with different population densities, and designing defense techniques.

6. ACKNOWLEDGMENTS

The authors wish to thank X. Sean Wang for his insightful comments on a preliminary version of this paper.

7. REFERENCES

[1] C. Bettini, S. Jajodia, P. Samarati, and X. S. Wang. *Privacy in Location-Based Applications*, volume 5599 of *LNSC*. Springer, 2009.

[2] C. U. Center for International Earth Science Information Network (CIESIN) and C. I. de Agricultura Tropical (CIAT). Gridded population of the world, version 3 (gpwv3), 2005.

[3] G. Ghinita. Private queries and trajectory anonymization: a dual perspective on location privacy. *Trans. Data Privacy*, 2(1):3–19, Apr. 2009.

[4] G. Ghinita, C. R. Vicente, N. Shang, and E. Bertino. Privacy-preserving matching of spatial datasets with protection against background knowledge. In *Proc. of the 18th SIGSPATIAL Int. Conf. on Advances in Geographic Information Systems*, GIS '10. ACM, 2010.

[5] K. Liu, C. Giannella, and H. Kargupta. An attacker's view of distance preserving maps for privacy preserving data mining. In *Proc. of the 10th Eur. Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, volume 4213. Springer, 2006.

[6] S. Mascetti, C. Bettini, and D. Freni. Longitude: Centralized privacy-preserving computation of users' proximity. In *Proc. of 6th VLDB workshop on Secure Data Management*, LNCS. Springer, 2009.

[7] R. Sibson. SLINK: an optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34, 1973.

[8] L. Šikšnys, J. R. Thomsen, S. Šaltenis, and M. L. Yiu. Private and flexible proximity detection in mobile social networks. In *Proc. of the 11th Int. Conf. on Mobile Data Management.* IEEE Comp. Soc., 2010.

[9] L. Šikšnys, J. R. Thomsen, S. Šaltenis, M. L. Yiu, and O. Andersen. A location privacy aware friend locator. In *Proc. of the 11th Int. Symposium on Spatial and Temporal Databases*, volume 5644 of *LNCS*. Springer, 2009.

[10] W. K. Wong, D. W.-l. Cheung, B. Kao, and N. Mamoulis. Secure knn computation on encrypted databases. In *Proc. of the 2009 ACM SIGMOD Int. Conf. on Management of data*, SIGMOD '09, pages 139–152. ACM, 2009.

[11] G. Zhong, I. Goldberg, and U. Hengartner. Louis, Lester and Pierre: Three protocols for location privacy. In *Privacy Enhancing Technologies*, volume 4776 of *LNSC*, pages 62–76. Springer, 2007.