Anonymizing Geo-Social Network Datasets

Amirreza Masoumzadeh and James Joshi
School of Information Sciences, University of Pittsburgh
135 N. Bellefield Ave., Pittsburgh, PA 15213, USA
[amirreza, jjoshi]@sis.pitt.edu

ABSTRACT

Geo-social networking systems, such as Foursquare and Facebook Places, where users perform interactions based on their self-reported locations are growing fast nowadays. The location-rich social network data collected in such systems could be of research interest for various purposes. However, such datasets are at the risk of user re-identification and consequently privacy violation of the involved users if they are not adequately anonymzied. In this paper, we study the problem of anonymizing a geo-social network dataset, based on adversarial knowledge on location information of its users. We introduce k-anonymity-based properties for guaranteeing anonymity based on location information, provide a realistic model of location data in geo-social networks, and propose corresponding anonymization algorithms. We also evaluate the proposed solutions using a synthetic GSN dataset.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Spatial databases and GIS*; K.4.1 [**Computers and Society**]: Public Policy Issues—*Privacy*

General Terms

Algorithms, Security

Keywords

Geo-social network, privacy, anonymizaion

1. INTRODUCTION

Advances in positioning technologies and proliferation of location-enabled mobile devices has recently given rise to Geo-Social Networks (GSNs). These systems, which are also referred to as Location-Based Social Networks (LBSNs), are a type of social networking systems that primarily focus on location of users and application related to that. Users provide their location to these systems, often using location-enabled mobile devices and interactions between users and these systems and among users take place relative to the provided location. Foursquare, Facebook Places, and Yelp are examples of such systems.

Study of social networks is of significant interest of both academia and industry community. And online social networking systems have made it possible to collect huge volume of social network data. However, publishing such datasets has its complications regarding users' privacy. The recent research literature on publishing social network datasets has shown effective ways to re-identify nodes of naively anonymized social networks, where only user identifiers are removed. The attacks on naively anonymized datasets range from using nodes' degree in a network as identifying signature [7, 6], to actively implanting nodes in a network [1], to using other publicly available social networks for de-anonymization purposes [11]. A GSN dataset can be more vulnerable to privacy attacks as an adversary can also leverage users' location information for re-identification purposes. Use of location information in re-identification has been well investigated in the context of location-based services, and various privacy preserving protocols and anonymization techniques has been suggested as possible solutions.

In this paper, we investigate an anonymization approach for GSN datasets, that considers both location and social connections. To be more specific, we consider datasets collected by systems such as Foursquare where users can befriend other users in the system, and check into location venues. Such a GSN dataset is essentially a social network of users, i.e., users and their relations with each other, and a series of logged locations for each of the users in the network. The log may contain specific location information and the times at which a user has reported those locations. Various privacy attacks can be launched against a naively anonymized GSN dataset. We focus on re-identification attack based on adversary's background knowledge about user locations. In a recent large-scale study of location data collected from cellphone users [15], Zang and Bolot report that a significant percentage of cellphone users are uniquely identifiable based on their top two or three locations. Moreover, Noulas et al. [12] report that home and corporate/office places are the top two locations from which people perform check-in in Foursquare. Motivated by these studies, we argue that check-in locations in a GSN dataset can be used by attackers to re-identify a target user. Moreover, the location information of the target's connections may strengthen the possibility of re-identification. For instance, an attacker

SPRINGL '11 November 1, 2011. Chicago, IL, USA
Copyright 2011 ACM ISBN 978-1-4503-1032-1/11/11 ...$10.00.

may know that the target frequently visits a certain coffee shop, and also knows about his/her workplace. In addition, the attacker may know the home address of a colleague of the target at work. Such background knowledge may easily enable re-identification of the target. We formulate the above problem and propose k-anonymization techniques to thwart such attacks.

In this work, unlike the literature on social network anonymization, we do not consider friendship structure of a target node as a feasible background knowledge for adversaries. That is mainly because our observations on GSNs such as Foursquare show that users on average have much smaller number of friends in these systems than general purpose SNSs such as Facebook. Therefore, their social connections is hardly representing their real friendship network. Instead, we focus on the location information revealed by social connections of a user that can assist in re-identification. Our contributions in this work can be summarized as follows:

- We formulate a simple and abstract model of GSNs, based on which we introduce anonymity notions for GSN datasets, i.e., $\mathcal{L}$-anonymity and $\mathcal{L}^2$-anonymity.
- We present an appropriate location model for GSNs, called *top m locations*, that we use as underlying data model for our algorithms.
- We propose algorithms for anonymizing a GSN dataset according to the notions of $\mathcal{L}$-anonymity and $\mathcal{L}^2$-anonymity.
- We present experimental results on running the proposed algorithms on a synthetic dataset, that we simulated based on published statistics about real cellphone users' location.

To the best of our knowledge, this is the first work to study anonymization of GSN datasets. However, as mentioned earlier, related work has studied anonymization in the context of location-based services and social networks, separately. The rest of the paper is structured as follows. In Section 2, we introduce the notions of location equivalence and corresponding anonymity properties for GSN datasets. Section 3 presents an appropriate location model for GSNs. In Section 4, we propose algorithms to anonymize a GSN dataset based on our proposed anonymity properties. We report results gained from running the algorithms on a synthetic dataset in Section 5. We survey briefly the related work in Section 6, and finally, conclude the paper pointing out future directions in Section 7.

2. K-ANONYMITY FOR GSN DATASETS

As mentioned earlier, removing explicit identifiers such as name from GSN datasets would not provide adequate privacy for users. An adversary might be able to re-identify a target user based on location information that exists about her. Such re-identification may lead to privacy breaches such as revealing exact location traces, friends, or other privacy-sensitive attributes existing in a GSN dataset. We formally define a geo-social network as follows:

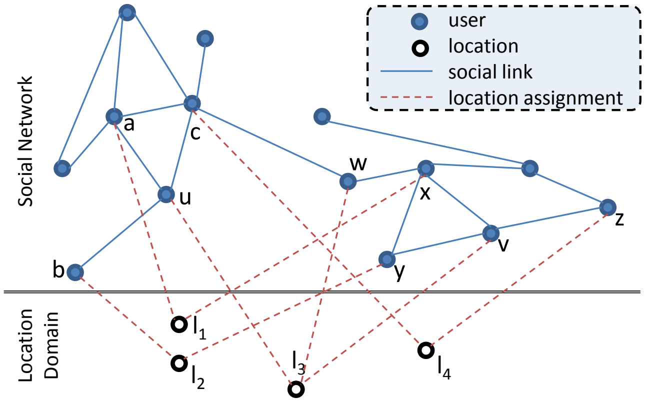

Figure 1: A small sample GSN

Definition 1. *A geo-social network is a 4-tuple $GSN = \langle V, E, L, \mathcal{L} \rangle$, where V is a set of users, $E \subseteq V \times V$ is a set of relationships between users, L is a domain for location information, and $\mathcal{L} : V \rightarrow L$ is a function that assigns location information to users.*

In the above definition, we consider the location domain to be abstract. It may refer to georaphic coordinates, street addresses, logical locations, or even more complex types, such as a number of top locations associated with a user. We will provide specific location model when proposing the anonymization problem. Figure 1 illustrates a small GSN dataset. Users that are connected via social links form a social network. Each user is also assigned a location value in the location domain. Since we are interested in anonymity based on location information, we introduce the following notion of location equivalence for GSN users.

Definition 2. *Users u and $v \in V$ are called $\mathcal{L}$-equivalent if they are assigned the same location information, i.e., $u \equiv_{\mathcal{L}} v \leftrightarrow \mathcal{L}(u) = \mathcal{L}(v)$.*

In Figure 1, users u, v, and w are $\mathcal{L}$-equivalent since all are assigned to location value l_3. Similarly, users a and x are $\mathcal{L}$-equivalent. Intuitively, anonymity can be provided by ensuring enough $\mathcal{L}$-equivalent users for every user in a GSN dataset. The following property captures the notion of k-anonymity based on $\mathcal{L}$-equivalence, which is called $\mathcal{L}_k$-*anonymity*.

Definition 3. *A GSN $\langle V, E, L, \mathcal{L} \rangle$ is $\mathcal{L}_k$-anonymous iff for every user $v \in V$, there are at least $k-1$ other users that are $\mathcal{L}$-equivalent to v. Formally, $\forall v \in V \exists v_1, v_2, \ldots, v_{k-1} \in V, v \equiv_{\mathcal{L}} v_1 \equiv_{\mathcal{L}} v_2 \ldots \equiv_{\mathcal{L}} v_{k-1}$.*

Based on the above property, an attacker cannot re-identify a user with a certainty greater than $1/k$ by knowing about the target's location information. $\mathcal{L}_k$-anonymity considers only a target's location information as background knowledge and no further information about social relationships of a target. However, for a GSN dataset, an attacker may leverage knowledge about location information of a target's friends to perform more accurate re-identification attacks. We introduce the following location equivalence relation that takes into account a user's friends' location information.

Definition 4. *Users u and v are $\mathcal{L}^2$-equivalent if, in addition to themselves, their adjacent users in the social network are $\mathcal{L}$-equivalent. Formally, $u \equiv_{\mathcal{L}^2} v \leftrightarrow \mathcal{L}(u) = \mathcal{L}(v) \wedge \{\mathcal{L}(u') | \langle u, u' \rangle \in E\} = \{\mathcal{L}(v') | \langle v, v' \rangle \in E\}$.*

In Figure 1, users u and v are $\mathcal{L}^2$-equivalent; they are $\mathcal{L}$-equivalent, and the set of u's friends' location information, i.e., $\{l_1, l_2, l_4\}$, is equal to v's. A k-anonymity property based on $\mathcal{L}^2$-equivalence is defined as follows.

Definition 5. *A GSN $\langle V, E, L, \mathcal{L} \rangle$ is $\mathcal{L}_k^2$-anonymous iff for every user $v \in V$, there are at least $k-1$ other users that are $\mathcal{L}^2$-equivalent to v. Formally, $\forall v \in V \exists v_1, v_2, \ldots, v_{k-1} \in V, v \equiv_{\mathcal{L}^2} v_1 \equiv_{\mathcal{L}^2} v_2 \ldots \equiv_{\mathcal{L}^2} v_{k-1}$.*

$\mathcal{L}_k^2$-anonymity protects against re-identification attacks based on a target's friends' locations. For instance, suppose an attacker knows that the target is working at the IS department in the University of Pittsburgh. Also, the attacker knows that she has a friend that works in the CS department and another friend that frequently visits a specific coffee shop. If such information is recorded as location information in a GSN dataset, $\mathcal{L}_k^2$-anonymity ensures that there are at least k users that may be matched against such background knowledge. $\mathcal{L}_k^2$-anonymity is obviously a stronger property than $\mathcal{L}_k$-anonymity, and consequently costlier to guarantee.

3. THE TOP M LOCATIONS MODEL

The anonymity properties defined in Section 2 are abstract with regards to the location model, i.e., no specific model is assumed. In this section, we introduce a parametric location model, called TL_m, the top m locations model. In the TL_m model, the location information consists of the top m locations that may act as a user's signature, assisting an attacker to re-identify her. These can be, for instance, the top locations based on frequency of a user's visit, which can be quite identifying for a user, according to the study in [15]. For geo-social networking systems such as Foursquare, the top frequent check-ins of a user may include workplace, a coffee shop on the way or close to work, etc. It is reasonable also to consider locations which are unique to a user even without frequent check-ins. Such locations may include target's home or her close friends' places. We formally define TL_m as follows.

Definition 6. *The TL_m location domain contains values as m-tuples such as $\langle r_1, r_2, \ldots, r_m \rangle$, where every r_i is a rectangular region.*

In the above definition, a *rectangular region* is a geographic area surrounded by a rectangle. Rectangular region r is represented using a 4-tuple $\langle x, y, w, h \rangle$, where x and y are geographic coordinates of the top-left corner, and w and h are width and height of the area, respectively. Zero width and height indicate region as a point. We use dot notation to represent members of a tuple, e.g., $l.r_1$ represents region r_1 belonging to location l.

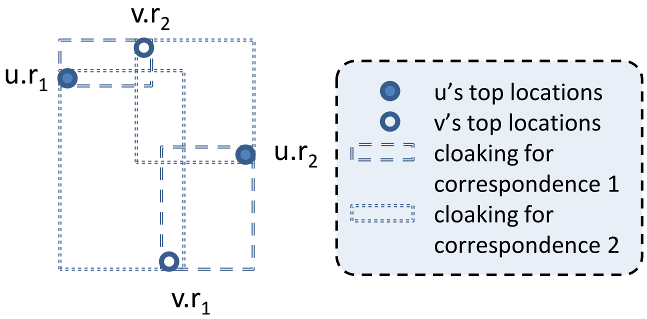

Figure 2: Example of Cloaking in TL_m Model

The TL_m model captures a realistic and reasonably powerful model of an adversary's background knowledge about a target's location information in GSNs. According to the data reported about sector location of cellphone users [15], more than 80% of them are uniquely identifiable based on their top three locations. Based on the top two locations, about 45% are uniquely identifiable, and more than 85% have at most one other user with the same location information. Note that a cellular sector is not of finer granularity than specific places people report, for instance, in Foursquare. Specific to GSNs, based on the data reported in [12], home and office locations are the two top places people report on average in Foursquare, which seems enough for their re-identification. The TL_m model is at the same time not so complex for anonymization purposes. We elaborate on this aspect in Section 4.

4. ANONYMIZATION ALGORITHMS FOR THE TOP M LOCATIONS MODEL

4.1 GSN $\mathcal{L}_k$-anonymization

We propose to use clustering algorithms to anonymize GSNs based on TL_m location model. The goal is to form users into clusters of size at least k; then report the same cloaked location information for the users in the same cluster. Cloaking multiple location points is usually performed by finding a minimum bounding rectangle that includes all the locations [10, 4]. The cloaking process in TL_m model is more complicated since each user's location consists of m regions. Consider users u and v's TL_2 locations depicted in Figure 2. Each user has r_1 and r_2 regions as a point. Combining the two location values results in a third location value with two regions. However, there are alternative ways for performing this. Each of the u's regions can be considered corresponding to any of v's regions for cloaking purpose. This results in two different *region correspondences*: $\{\langle \mathcal{L}(u).r_1, \mathcal{L}(v).r_2 \rangle, \langle \mathcal{L}(u).r_2, \mathcal{L}(v).r_1 \rangle\}$ and $\{\langle \mathcal{L}(u).r_1, \mathcal{L}(v).r_1 \rangle, \langle \mathcal{L}(u).r_2, \mathcal{L}(v).r_2 \rangle\}$. As shown in Figure 2, the former correspondence results in smaller cloaked areas for the results than the latter. Therefore, it is a better option in terms of preserving location accuracy of the original data points. The correspondence that results in the smallest cloaking among the alternatives can be used as a natural distance measure for our clustering approach. Location information of two nodes can be combined to form larger cluster of nodes that meet anonymity property, while minimum cloaking is applied to preserve the location information at best.

We now define this distance measure. When two TL_m values are considered to be combined in one cluster, corresponding regions in the two values should be combined (cloaked) into one region. Since there are m different regions in each location value, there will be $m!$ different combinations for region correspondences. As mentioned earlier, we consider the minimum expansion in area that results from cloaking based on any of such correspondences as the distance measure between two TL_m values. The following definition formally captures the notion of distance in TL_m. Operator $\otimes$ in the following definition is an element-by-element binary operator for equally sized tuples, which outputs pairs of elements of first and second operands. For instance, $\langle A, B, C\rangle \otimes \langle 1, 2, 3\rangle = \{\langle A, 1\rangle, \langle A, 2\rangle, \langle A, 3\rangle\}$. Also, function P calculates the set of all permutation tuples for a given set. For instance, $P(\{1, 2, 3\}) = \{\langle 1, 2, 3\rangle, \langle 1, 3, 2\rangle, \langle 2, 1, 3\rangle, \langle 2, 3, 1\rangle, \langle 3, 1, 2\rangle, \langle 3, 2, 1\rangle\}$.

Definition 7. *The distance between TL_m values t and s is calculated by the following formula:*

$$D(t, s) = \min_{C \in P(\langle 1,\ldots,m\rangle)} \sum_{\langle i,j\rangle \in \langle 1,\ldots,m\rangle \otimes C} MBRA(t.r_i, s.r_j)$$

where $MBRA$ calculates the area of minimum bounding rectangle of two regions.

In the above definition, a correspondence is formed based on each permutation of numbers $1, \ldots, m$, and the sum of the minimum bounding rectangle areas of the corresponding regions are calculated. The minimum value among the calculated values for all correspondences is selected as the distance between the two locations. In the example depicted in Figure 2, $MBRA$ calculates the area of each dashed rectangle. It is easy to see that the sum of areas of the bounding rectangles associated with correspondence 1 is smaller than those for correspondence 2. Therefore, the sum of the areas of the line-dashed rectangles is regarded as the distance between the locations of u and v.

Given the above distance measure, we follow a clustering approach similar to the Union-Split method [13], to create clusters with minimum k nodes, and report an anonymized value for each cluster. However, we modify the algorithm to accommodate our data model, which is significantly different than the original numerical values it was proposed for. Algorithm 1 shows the pseudo code for $\mathcal{L}_k$-anonymization. It is a special hierarchical agglomerative clustering algorithm that stops when each cluster has at least k members. The algorithm starts with each user as a separate cluster, with the cluster center set as her location. In each iteration, the distance between every pair of clusters is computed. The pair of clusters with the minimum distance are selected to be merged. The cluster center of the merger is set as the minimum bounding rectangle of the correspondence between the two clusters that results in the minimum distance. If the merger has more than two times the desired size of anonymity clusters, i.e., more than $2k$, the cluster needs to be broken into two (preferably equally-sized) clusters. We perform this to avoid ending up having clusters significantly larger than the required anonymity size. The larger the size of an anonymity set, the more probable the location distortion due to cloaking. One difference between our approach and the one proposed in [13], is the use of a *k-medoid clustering* approach ($k = 2$) for splitting (performed at step 8) instead of *k-means clustering.*

Algorithm 1 $\mathcal{L}_k - anonymize$

Input: GSN dataset $GSN = \langle V, E, TL_m, \mathcal{L}\rangle$, where $|V| = n$, and the anonymization parameter k.
Output: $\mathcal{L}_k$-anonymized dataset $GSN' = \langle V, E, TL_m, \mathcal{L}'\rangle$.

```
 1: Initialize $C$ as $\{c_j | 1 \leq j \leq n, c_j.users = \{v_j \in V\}, c_j.center = \mathcal{L}(v_j)\}$
 2: while $\exists c \in C, |c.users| < k$ do
 3:   for all $c_i, c_j \in C$ do
 4:     Calculate distance $D(c_i.center, c_j.center)$ according to Definition 7
 5:   end for
 6:   Merge $c_x$ and $c_y$ into $c_m$, where $D(c_x.center, c_y.center)$ is minimum
 7:   if $|c_m.users| \geq 2k$ then
 8:     Split $c_m$ into $c_{m1}$ and $c_{m2}$
 9:   end if
10:   Update $C$
11: end while
12: $GSN' \leftarrow GSN$
13: for all $c \in C$ do
14:   for all $u \in c.users$ do
15:     $\mathcal{L}'(u) \leftarrow c.center$
16:   end for
17: end for
18: return $GSN'$
```

Theorem 1. *Algorithm 1 outputs an $\mathcal{L}_k$-anonymous dataset as per Definition 3.*

Proof. All the users are members of clusters since the algorithm starts with every user as a single cluster and merges them iteratively. Also, the main loop in the algorithm (steps 2-11) does not terminate until all the clusters have at least k members, i.e., $\forall c \in C, |c.users| \geq k$. Therefore, assigning the same location information to all the members of the same cluster in steps 13-17 ensures that for every user v in a cluster c there are at least $k-1$ other users $v_1, v_2, \ldots, v_{k-1} \in c.users$ where $v \equiv_{\mathcal{L}} v_1 \equiv_{\mathcal{L}} v_2 \ldots \equiv_{\mathcal{L}} v_{k-1}$. □

The time complexity of an optimized implementation of Algorithm 1 is $O(m!n^2 \log n)$, where m is the parameter of the TL_m model, and n is the number of users. The m value is not expected to be more than 3 as a reasonable background knowledge. An optimized implementation can be achieved by maintaining a sorted list of distances for each cluster, and updating only the entries related to the merged cluster at each iteration ($O(n \log n)$). With the assumption of rare need for split, the algorithm needs n iterations at most to merge the clusters.

4.2 GSN $\mathcal{L}_k^2$-anonymization

We build on our proposed algorithm in Section 4.1 to anonymize a GSN dataset based on Definition 5. In fact, we rely on the clustering and cloaking performed by the $\mathcal{L}_k$-anonymization algorithm. Since Algorithm 1 clusters users and make users in the same cluster $\mathcal{L}_k$-equivalent, we only modify edges in the social network to ensure the $\mathcal{L}_k$-equivalence of neighbors of users in each cluster. We take two different approaches for performing this. In the first approach, the edges in the original network are preserved and only new edges are inserted. Therefore, no edge information in the social network is lost during anonymization. In the second approach, edges are both inserted and removed to assure the anonymity property. The rationale behind this approach is to avoid too much increase in the size of the social network due to anonymization.

Algorithm 2 shows the pseudo code for our *insert-only* approach. We fist perform the steps in Algorithm 1 to achieve the resultant clusters and the $\mathcal{L}_k$-anonymous dataset. Next, for any two clusters c_i and c_j, we form set E_{inter} of inter-cluster edges between the clusters (edges connecting a member from one to a member from the other). If the set is not empty, we ensure that all the users in one cluster have at least a neighbor in the other cluster. If a user needs to have a neighbor from another cluster, one of the members of that cluster is randomly chosen to be connected. Note that c_i and c_j can refer to the same cluster in the special case. In that case, the edges are intra-cluster instead of inter-cluster. However, the same procedure applies to guarantee $\mathcal{L}_k$-equivalent neighbors for the users.

Algorithm 2 $\mathcal{L}_k^2$-iAnonymize (insert-only)

Input: GSN dataset $GSN = \langle V, E, TL_m, \mathcal{L}\rangle$, where $|V| = n$, and the anonymization parameter k.
Output: $\mathcal{L}_k^2$-anonymized dataset $GSN' = \langle V, E, TL_m, \mathcal{L}'\rangle$.

```
1: $\mathcal{L}_k - anonymize(GSN)$
2: for all $c_i, c_j \in C$ do
3:    $E_{inter} \leftarrow \{\langle u, v\rangle \in E | \exists u \in c_i.users, v \in c_j.users\}$
4:    if $|E_{inter}| > 0$ then
5:       for all $u \in c_i.users$ do
6:          if $\nexists v \in c_j.users, \langle u, v\rangle \in E$ then
7:             $E' \leftarrow E'\langle u, v\rangle \in E$, where $v \in c_j.users$ is randomly chosen
8:          end if
9:       end for
10:      for all $v \in c_j.users$ do
11:         if $\nexists u \in c_i.users, \langle u, v\rangle \in E$ then
12:            $E' \leftarrow E'\langle u, v\rangle \in E$, where $u \in c_i.users$ is randomly chosen
13:         end if
14:      end for
15:   end if
16: end for
17: return $GSN'$
```

In the following, we prove correctness of the proposed algorithm for $\mathcal{L}_k^2$-anonymization.

THEOREM 2. *Algorithm 2 outputs an $\mathcal{L}_k^2$-anonymous dataset as per Definition 5.*

PROOF. The first step of the algorithm generates an $\mathcal{L}_k$-anonymous dataset, and corresponding clusters with $\mathcal{L}$-equivalent members, according to Theorem 1. We need to show that every member of a cluster is also $\mathcal{L}^2$-equivalent to other members of the same cluster. Consider an arbitrary cluster c and one of its members $v \in c.users$. For any other cluster member $u \neq v \in c.users$, $u \equiv_{\mathcal{L}} v$ according to Theorem 1. Also, for any edge adjacent to v, say $\langle v, v'\rangle \in E'$, there exists at least an adjacent edge to u, say $\langle u, u'\rangle \in E'$, where v' and u' are in the same cluster. This was assured by inserting the edges in the social network in steps 2-16. Therefore, set $\{\mathcal{L}(u') | \langle u, u'\rangle \in E'\}$ will be equal to set $\{\mathcal{L}(v') | \langle v, v'\rangle \in E'\}$, which completes the proof for $u \equiv_{\mathcal{L}^2} v$. □

The time complexity of Algorithm 2 is bounded by the time complexity of Algorithm 1, i.e., $O(m!n^2 \log n)$. This is because the main loop in Algorithm 2 has time complexity $O(n^2)$. Considering the minimum cluster size k, there will be at most $\lfloor n/k \rfloor$ clusters at the end, and therefore $O(n^2/k^2)$ different pairs of clusters. Each cluster has less than $2k$ nodes, due to the splitting mechanism. So enumerating members of every pair of clusters for edge insertion purpose has complexity $O(k^2)$. Therefore, the overall complexity of the main loop is $O(n^2/k^2) \times O(k^2) = O(n^2)$.

Algorithm 3 $\mathcal{L}_k^2$-irAnonymize (insert/remove)

Input: GSN dataset $GSN = \langle V, E, TL_m, \mathcal{L}\rangle$, where $|V| = n$, and the anonymization parameter k.
Output: $\mathcal{L}_k^2$-anonymized dataset $GSN' = \langle V, E, TL_m, \mathcal{L}'\rangle$.

```
$\mathcal{L}_k - anonymize(GSN)$
for all $c_i, c_j \in C$ do
   $E_{inter} \leftarrow \{\langle u, v\rangle \in E | \exists u \in c_i.users, v \in c_j.users\}$
   if $|E_{inter}| < \theta$ then
      $E' \leftarrow E' \setminus E_{inter}$
   else
      for all $u \in c_i.users$ do
         if $\nexists v \in c_j.users, \langle u, v\rangle \in E$ then
            $E' \leftarrow E'\langle u, v\rangle \in E$, where $v \in c_j.users$ is randomly chosen
         end if
      end for
      for all $v \in c_j.users$ do
         if $\nexists u \in c_i.users, \langle u, v\rangle \in E$ then
            $E' \leftarrow E'\langle u, v\rangle \in E$, where $u \in c_i.users$ is randomly chosen
         end if
      end for
   end if
end for
return $GSN'$
```

As mentioned earlier, in the second approach for $\mathcal{L}_k^2$-anonymization, we perform both edge insertion and removal, as depicted in Algorithm 3. The algorithm performs similar to Algorithm 1, except that it removes the inter-cluster edge set E_{inter} from the network if the size of the set is less than a threshold θ. Otherwise, it continues with inserting corresponding inter-cluster edges for every member of the

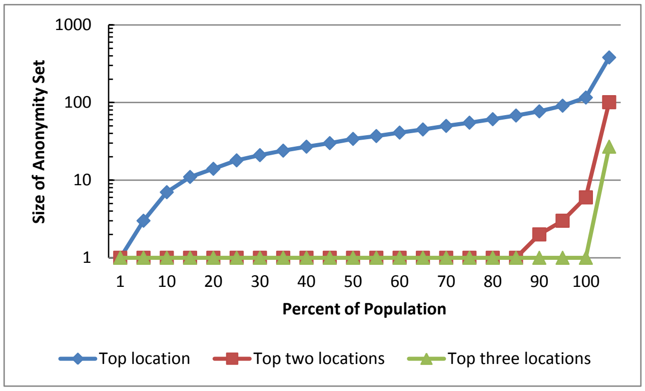

Figure 3: Size of the anonymity sets in the dataset when top 1, 2, or 3 locations are revealed

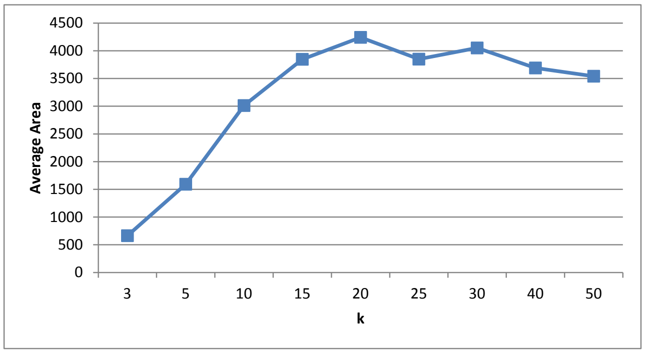

Figure 4: Average area size for cloaked regions for different k values in $\mathcal{L}_k$-anonymization

two clusters. We set θ to be half of the size of the smaller cluster between c_i and c_j. Algorithm 3 has the same time complexity as Algorithm 2, i.e., $O(m!n^2 \log n)$. We omit the correctness proof for the algorithm, which is quite similar to the proof of Theorem 2.

5. EXPERIMENTAL RESULTS

Given limitations to access a real GSN dataset, we generated a synthetic GSN dataset to evaluate the proposed anonymization algorithms. We used TL_3 as our location model, i.e., each user is assigned with her top three locations. In the generation process, we used the statistics reported in [15] on a nation-wide cellular network in order to have realistic distributions regarding identifiability of users with regards to their locations. More specifically, we leveraged the distribution of anonymity group sizes with regards to considering one, two, and three top locations per cellular sector. However, we scaled down the dataset size to 1500 users (the data reported in [15] is about 25 million users). Figure 3 shows the size of anonymity sets for different percentile of users in our dataset, depending on the condition of revealing one, two, or three locations. In our synthetic dataset, each of the three user's regions is a point, randomly chosen on a 1000 by 1000 unit square-shaped area. These points are converted to cloaked regions as the result of the anonymization algorithm.

Figure 4 depicts the performance of $\mathcal{L}_k$-anonymization, showing average area of the regions in users' location information as a result of choosing different k values. As expected, the area has an increasing trend with the increase of k. However, we notice that the regions do not grow larger after $k = 20$ in our dataset.

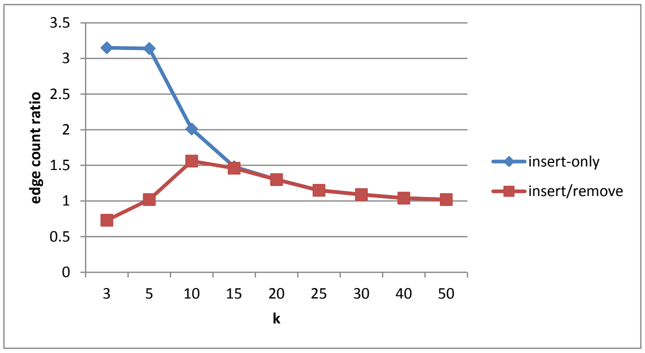

Figure 5: Edge count ratio of the $\mathcal{L}_k^2$-anonymized to the original dataset for different k values

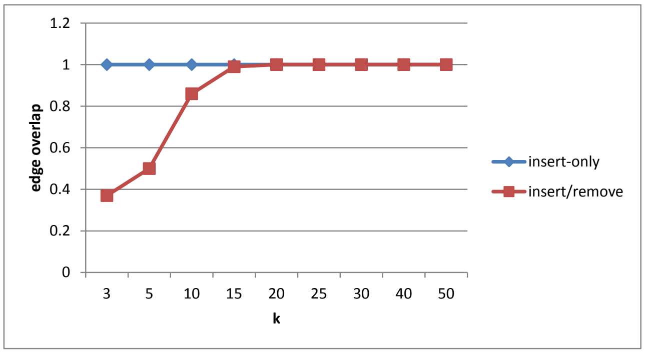

Figure 6: Edge overlap of the $\mathcal{L}_k^2$-anonymized with the original dataset for different k values

For $\mathcal{L}_k^2$-anonymization, we experimented with both insert-only and insert/remove approaches on our dataset. Note that since our $\mathcal{L}_k^2$-anonymization algorithm is based on clusters created by the $\mathcal{L}_k$-anonymization algorithm, the same results apply about cloaking performance. Figures 5 and 6 show the edge count ratio and overlap of the result of $\mathcal{L}_k^2$-anonymization algorithm compared to the original dataset for different k values, respectively. The insert-only approach tends to insert a lot of edges into the social network for lower k values (e.g., more than 2 times of the original size at $k = 5$). Comparatively, the add/remove approach introduces less variation to the edge count. Regarding maintaining the original edges of the social network, the insert-only approach has the obvious advantage of keeping the original edges intact. But the insert/remove approach provides less overlap for low k values, and it improves with increase in k. In fact, both approaches seem to perform similar at larger k values, i.e., $k = 15$ or higher, both in terms of edge count ratio and overlap.

6. RELATED WORK

Various techniques have been proposed based on location cloaking, i.e., reporting a larger area rather than a user's exact location, in order to provide k-anonymity for location-based service users. Approaches such as *New Casper* [10], *Privé* [5], and *PrivacyGrid* [2] cloak each query's location to include at least k other users. Therefore, an attacker's certainty about a specific user's issuance of a query is at most $1/k$. Other approaches such as *CliqueCloak* [4] collect and submit k queries with the same cloaked location at the same time to an LBS. The problem in LBS anonymization techniques is slightly different than the problem discussed in this paper as it deals with anonymizing queries one at a time. In contrast, anonymizing a location-contained dataset involves more rigorous optimization as it needs to anonymize all the records at the same time. Moreover, our anonymization technique deals with a more complex location model, i.e., top m locations, rather than a single location for each user. Another obvious contrast is consideration network connections in our approach.

Re-identification attacks on social network datasets and anonymization techniques to prevent them have been a hot research topic recently. Backstrom et al. present a family of active/passive attacks that work based on uniqueness of some small random subgraphs embedded in a network [1]. Hay et al. show significantly low k-anonymity in real, naively anonymized social networks when considering structural queries such as degree of a target node as adversarial background knowledge [7, 6]. The proposed social network anonymization approaches in the literature can be categorized into two groups: graph generalization and graph perturbation. In *generalization techniques* [6, 16, 3], the network is first partitioned into subgraphs. Then each subgraph is replaced by a supernode, and only some structural properties of the subgraph alongside linkage between clusters are reported. In *perturbation techniques* , the network is modified to meet desired privacy requirements. This is usually carried out by adding and/or removing graph edges. The perturbation methods include randomly adding/ removing edges [7, 14], and providing k-anonymity in terms of node degrees [8, 13] and node neighborhood [17]. Naturally, the focus of social network anonymization approaches is on anonymizing structural patterns such as node degree and neighborhood, and not on information associated with the nodes. However, some approaches such as [17] also consider anonymizing node labels based on generalization trees. Nevertheless, none of these methods deal with location data associated with nodes as needed for anonymization of GSN datasets.

7. CONCLUSIONS AND FUTURE WORK

Geo-social networks are growing fast and becoming popular social networking tools, which naturally brings up the privacy issues with regards to the huge amount of the location-rich data collected by these systems. Proper anonymization mechanisms are necessary to be developed specifically for GSN datasets since the location information is more complex and harder to anonymize and preserve at the same time, compared to other simpler data attributes.

In this work, we formalized and proposed two notions of anonymity with regards to location information in GSN datasets, namely, $\mathcal{L}_k$-anonymity and $\mathcal{L}_k^2$-anonymity. Also, we proposed algorithms for anonymizing a GSN dataset based on these notions, and reported some experimental results on their performance.

In our approach to $\mathcal{L}_k^2$-anonymization, we chose to rely on our $\mathcal{L}_k$-anonymization algorithm for clustering purpose, and perform edge perturbation to satisfy the required location equivalence condition for users' neighbors. An alternative approach may be to perform both tasks at the same time, i.e., looking for clusters that minimize both the cloaking area and distortion to the network structure to satisfy the anonymity property. In particular, approaches to preserve network structure in anonymization [9] can be useful in this regard. We leave this approach for future work. We performed our experiments on a synthetic GSN dataset, due to the limitations to obtain a real dataset. Although in our data generation method we tried to replicate anonymity characteristics of a real dataset, it cannot fully represent the properties of a real GSN dataset due to the unknown parameters and randomness assumed. As a future work, we plan to collect a real dataset from Foursquare system to further evaluate the feasibility and performance of our anonymization approaches. Moreover, we will consider probably more generic location models suitable for anonymization of GSN datasets.

Acknowledgment

This research has been supported by the US National Science Foundation award IIS-0545912.

8. REFERENCES

[1] L. Backstrom, C. Dwork, and J. Kleinberg. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *Proceedings of the 16th international conference on World Wide Web*, WWW '07, pages 181–190, New York, NY, USA, 2007. ACM.

[2] B. Bamba, L. Liu, P. Pesti, and T. Wang. Supporting anonymous location queries in mobile environments with PrivacyGrid. In *Proc. 17th Int'l Conference on World Wide Web*, pages 237–246. ACM, 2008.

[3] A. Campan and T. M. Truta. A clustering approach for data and structural anonymity in social networks. In *Proceedings of the 2nd ACM SIGKDD International Workshop on Privacy, Security, and Trust in KDD (PinKDD'08), in Conjunction with KDD'08*, 2008.

[4] B. Gedik and L. Liu. Protecting Location Privacy with Personalized k-Anonymity: Architecture and Algorithms. *IEEE Transactions on Mobile Computing*, 7(1):1–18, 2008.

[5] G. Ghinita, P. Kalnis, and S. Skiadopoulos. PRIVE: anonymous location-based queries in distributed mobile systems. In *Proc. 16th Int'l Conference on World Wide Web*, pages 371–380, New York, NY, USA, 2007. ACM.

[6] M. Hay, G. Miklau, D. Jensen, D. Towsley, and P. Weis. Resisting structural re-identification in anonymized social networks. *Proc. VLDB Endow.*, 1(1):102–114, Aug. 2008.

[7] M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava. Anonymizing Social Networks. Technical Report 07-19, University of Massachusetts Amherst, 2007.

[8] K. Liu and E. Terzi. Towards identity anonymization on graphs. In *SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 93–106, New York, NY,

USA, 2008. ACM.

[9] A. Masoumzadeh and J. Joshi. Preserving Structural Properties in Anonymization of Social Networks. In *Proc. 6th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2010)*, Oct. 2010.

[10] M. F. Mokbel, C. Y. Chow, and W. G. Aref. The new Casper: Query processing for location services without compromising privacy. In *Proc. 32nd Int'l Conference on Very Large Data Bases*, pages 763–774. ACM, Sept. 2006.

[11] A. Narayanan and V. Shmatikov. De-anonymizing Social Networks. *Security and Privacy, IEEE Symposium on*, 0:173–187, Aug. 2009.

[12] A. Noulas, S. Scellato, C. Mascolo, and M. Pontil. An empirical study of geographic user activity patterns in foursquare. In *Proc. of the 5th Int'l AAAI Conference on Weblogs and Social Media*, pages 570–573, 2011.

[13] B. Thompson and D. Yao. The union-split algorithm and cluster-based anonymization of social networks. In *ASIACCS '09: Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, pages 218–227, New York, NY, USA, 2009. ACM.

[14] X. Ying, K. Pan, X. Wu, and L. Guo. Comparisons of randomization and K-degree anonymization schemes for privacy preserving social network publishing. In *SNA-KDD '09: Proceedings of the 3rd Workshop on Social Network Mining and Analysis*, pages 1–10, New York, NY, USA, 2009. ACM.

[15] H. Zang and J. C. Bolot. Anonymization of Location Data Does Not Work: A Large-Scale Measurement Study. In *Proc. of ACM Mobicom*, Sept. 2011.

[16] E. Zheleva and L. Getoor. Preserving the Privacy of Sensitive Relationships in Graph Data. In F. Bonchi, E. Ferrari, B. Malin, and Y. Saygin, editors, *Privacy, Security, and Trust in KDD*, volume 4890 of *Lecture Notes in Computer Science*, chapter 9, pages 153–171. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[17] B. Zhou and J. Pei. Preserving Privacy in Social Networks Against Neighborhood Attacks. In *2008 IEEE 24th International Conference on Data Engineering*, pages 506–515. IEEE, Apr. 2008.