Preserving Communities in Anonymized Social Networks

Alina Campan*, Yasmeen Alufaisan** and Traian Marius Truta*
* Department of Computer Science, Northern Kentucky University, Highland Heights, KY 41099, USA
E-mail: **{campana1, trutat1}@nku.edu**
** Department of Computer Science, The University of Texas at Dallas, Richardson, TX 75080, USA
E-mail: **yxa130630@utdallas.edu**

Abstract. Social media and social networks are embedded in our society to a point that could not have been imagined only ten years ago. Facebook, LinkedIn, and Twitter are already well known social networks that have a large audience in all age groups. The amount of data that those social sites gather from their users is continually increasing and this data is very valuable for marketing, research, and various other purposes. At the same time, this data usually contain a significant amount of sensitive information which should be protected against unauthorized disclosure. To protect the privacy of individuals, this data must be anonymized such that the risk of re-identification of specific individuals is very low. In this paper we study if anonymized social networks preserve existing communities from the original social networks. To perform this study, we introduce two approaches to measure the community preservation between the initial network and its anonymized version. In the first approach we simply count how many nodes from the original communities remained in the same community after the processes of anonymization and de-anonymization. In the second approach we consider the community preservation for each node individually. Specifically, for each node, we compare the original and final communities to which the node belongs. To anonymize social networks we use two models, namely, *k*-anonymity for social networks and *k*-degree anonymity. To determine communities in social networks we use an existing community detection algorithm based on modularity quality function. Our experiments on publically available datasets show that anonymized social networks satisfactorily preserve the community structure of their original networks.

1 Introduction

Social media and social networks are embedded in our society to a point that could not have been imagined only ten years ago. Facebook, LinkedIn, and Twitter are already well known social networks that have a large audience in all age groups. Recently more trendy social sites such as Pinterest, Instagram, Vine, Tumblr, WhatsApp, and Snapchat are being preferred by the younger audience [26]. The amount of data that those social sites gather from their users is continually increasing and this data is very valuable for marketing, research, and various other purposes. At the same time, this data usually contain a significant amount of sensitive information which should be protected against unauthorized disclosure. The above social sites treat seriously the privacy of their members and they provide a series of privacy controls and a privacy policy regarding of how the collected data is used. First, the privacy controls allow individuals to set up their privacy preferences/settings. Using these settings, a user may choose what personal information is available to each group of friends or what personal information is available to everyone on the internet. Second, the privacy policy lists how the social site will use the data from their users and how this data can be shared with third party companies such as advertising companies, etc. To protect the privacy of individuals, this data must be anonymized such that the risk of re-identification of specific individuals is very low.

In this paper we focus only on social network data model, which is one of the most common data models used in social media. The social network data (also referred as graph data or simply network data) should be made anonymous before being released in order to protect the privacy of individuals that are included in this social network. Due to a wide variety of problem assumptions, a standard social network anonymization model does not exist. One important assumption is what constitutes sensitive information which needs to be protected against disclosure. In general, either identity of individuals, their relationship, and/or part of their social network node content is considered sensitive [18]. A second aspect of anonymization is what anonymization approach is more appropriate to follow, and there are three choices that are analyzed in the literature: anonymization via clustering, anonymization via graph modification, and a hybrid approach [3, 7, 37, 39]. Considering these choices, it is not a surprise that the resulting anonymized networks are very dissimilar in terms of structure and in terms of preserving the original graph properties. In this paper we consider only the identity of individuals being sensitive information and we analyze two anonymization models. These models are: *k-anonymity for social networks* [7], a model from the anonymization by clustering family, which can be enforced on a network by using the *SaNGreeA* algorithm [7], and *k-degree anonymity* [18], a graph modification approach, enforced by the *Fast K-Degree Anonymization* (*FKDA*) algorithm [19].

The purpose of this work is to study whether anonymized social network preserve existing communities from the original social networks. Communities (also known as clusters) are groups of nodes from a social network which likely have similar proprieties or characteristics [12] Community detection is well studied in the literature and many different community detection algorithms have been presented in social network analysis literature. A good survey of these algorithms can be found in [12]. For this paper we focus on a specific community detection method known as Louvain method [4, 27], which is a heuristic algorithm based on modularity optimization [23]. The modularity is a quality function that can be computed for a graph partitioned in communities. Modularity has received a wide attention in recent years being used as a quality function in many community detection algorithms, to assess the stability of partitions [21], in determining graph visualization layouts [24], and in graph summarization [2].

To study how communities are preserved in anonymized social networks we follow several steps. First, we anonymize several real social networks using *SaNGreeA* and *Fast K-Degree Anonymization* algorithms. Second, we de-anonymize networks masked with *SaNGreeA* to allow fair comparison between the original and the anonymized network (details will be provided later). And third we use Louvain community detection algorithm to compare how well the communities are preserved between the original networks and their anonymized (via *Fast K-Degree Anonymization*) and de-anonymized *SaNGreeA* versions.

The contributions of this paper are as follows. First, to our knowledge, this is the first work that studies the community preservation in the context of anonymized social networks. Second, while we use specific anonymization models (*k*-anonymity for social networks and *k*-degree anonymity) and one community detection algorithm (Louvain) in our analysis, the proposed workflow is easily extensible to other social network anonymity models and community detection algorithms. And, third, we introduce in this paper two approaches to measure the community preservation between two networks (the initial network and its anonymized version). In the first approach named *naive community preservation,* we simply count how many nodes from the original communities remained in the same community after the processes of anonymization and de-anonymization. In the second measure named *community preservation at node level*, we consider the community preservation for each node individually. Specifically, for each node, we compare the original and final communities to which the node belongs.

The remaining of this paper is structured as follows. Section 2 presents related work. Section 3 describes the anonymity models used in this paper. Section 4 presents the de-anonymization models that we used with the anonymization via clustering networks. Section 5 describes the modularity function, the community detection algorithm used in this paper, and the two new measures to compute the community preservation. Section 6 contains the description of the workflow we used for our experiments, the description of the social network datasets, and the experimental results. Section 7 summarizes our conclusions.

2 Related Work

This paper applies several new findings in data privacy, social network analysis, and graph generators in a new more practical problem. To our knowledge this is the first paper that addresses how well the existing communities in social networks are preserved when these social networks are anonymized.

Related to this work are a series of papers that analyses the usefulness of anonymized social network for other social analysis tasks. Most of the previous works compare how well structural properties (diameter [14], centrality measures [13], clustering coefficients [33, 34] and/or topological indices [20]) are preserved between the original social networks and their anonymized versions. Three such papers considers anonymization via clustering in their analysis and they differs in which structural property are analyzed and how the anonymization/de-anonymization is performed [1, 31, 32]. Specifically, in [31], the preservation of radius, diameter, and centrality measures is investigated between the original social networks and the anonymized social networks. There was no de-anonymization technique performed in this paper. In [32], a random-based de-anonymization technique was introduced in order to better compare the structural properties between original and anonymized social networks. In addition to this novel approach, topological indices and clustering coefficients are also investigated. In [1], a more enhanced de-anonymization approach was introduces. In this approach an algorithm (RMAT [9]) that reconstructs a power-law network is used. In addition to this improvement, this work also compares the preservation of centrality measures between a cluster-based anonymization approach and a graph modification approach.

Other papers that discuss structural property preservation focus on how specific graph modification approaches (*k*-automorphism [29], *k*-isomorphism [11], and *k*-symmetry [35]) preserve a subset of those structural properties. In other related work, comparison of the most influential nodes and the spread of influence in social networks were performed between the original social networks and the anonymized/de-anonymized networks [8].

As already mentioned, related to this work are social network anonymization models, community detection in social networks, and graph generators models. Each of these topics is well covered in research literature. A good survey of existing social network anonymization models as well as other issues regarding privacy in social networks is covered in [38]. Various community detection techniques are also well studied in the literature [12, 17]. A survey of graph generators models is presented in [10]. In this paper we use the *Erdos-Renyi* random network model [5] and *R-MAT* power law model [9].

3 Social Network Anonymity Models

In this section the two anonymity models used in this paper, *k*-anonymity for social networks and *k*-degree anonymity, are presented. Since in this paper our focus is on community preservation based on the social networks structure, we make the additional simplifying assumption that the nodes in the social network do not have quasi-identifier attributes (such as *Age* and *ZipCode*); accordingly, the anonymization process is based on the social network structure only. Sensitive attribute values that need to be protected from potential intruders (such as *ICD9Code* and *Income*) are preserved in the social network.

Consider an initial social network modeled as a simple undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{E}$ is the set of edges. Only binary relationships are allowed in this model. Additionally, all relationships are of the same type and they are represented as unlabeled undirected edges. These edges are assumed to be known by an intruder. Based on this graph structure, an intruder is able to identify individuals and to reveal their sensitive information due to the uniqueness of their neighborhoods.

We illustrate an example of social network, labeled $\mathcal{G}_1$, in Figure 1. This network has 12 nodes and 12 edges (the nodes' colors will be used in the next subsection).

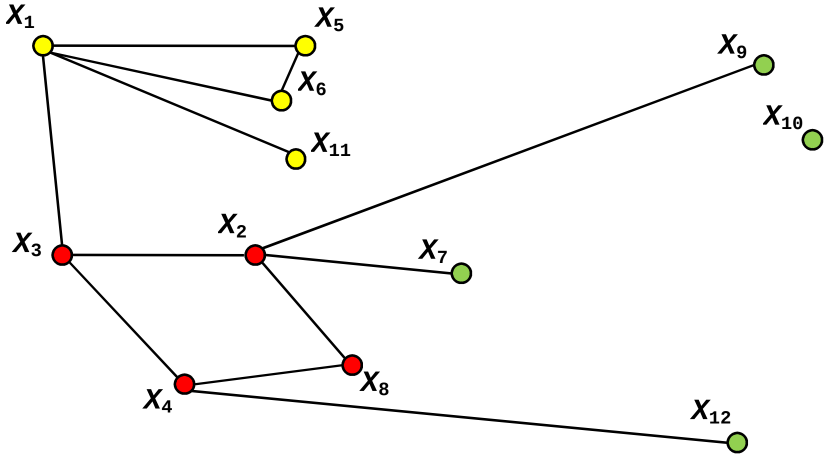

Figure 1. Social network example, $\mathcal{G}_1$.

3.1 K-Anonymity for Social Networks

In this model, the nodes from the social network are partitioned into pairwise disjoint clusters based on a similarity criteria. These clusters are generalized to super-nodes, which may be connected by super-edges. The goal of this process is to make any two nodes belonging to the same cluster indistinguishable based on their relationships. To achieve this objective, Campan and Truta developed intra-cluster and inter-cluster edge generalization techniques that were used for creating super-nodes and super-edges [7]. To satisfy the *k*-anonymity for social networks clustered model – model derived from the well-known *k*-anonymity property for microdata [28, 30], each cluster must have at least *k* nodes.

In the anonymized network, each cluster is replaced by a super-node and edges from the original network are generalized via an edge generalization process which preserves the number of edges, in other words, it does not add or delete edges. The edge generalization process is divided into two components: edge intra-cluster generalization and edge inter-cluster generalization.

Edge intra-cluster generalization is a process in which each of the clusters is generalized into a single super-node and the information released with it is the pair of values ($|cl|$, $|\mathcal{E}_{cl}|$), where $|X|$ represents the cardinality of the set X, cl represents the set of nodes in the cluster, and $\mathcal{E}_{cl}$ represents the set of edges that connect two nodes from *cl*. An example of such super-node information would be (4, 3), which means that the cluster has four of the original nodes with three edge between them. Hiding the precise connectivity information between nodes in the same cluster will protect the identity of cluster's nodes.

Edge inter-cluster generalization is a similar process for edges between two clusters. In the anonymized graph, the set of inter-cluster edges between any two clusters is generalized into one single super-edge. The information released due to this process is the value $|\mathcal{E}_{cl1,\ cl2}|$, where cl_1 and cl_2 are the two clusters and $\mathcal{E}_{cl1,\ cl2}$ represents the set of edges that connect the two clusters. In other words, each super-edge is described by the number of edges connecting nodes within the two super-nodes.

The algorithm used in the anonymization process, called the *SaNGreeA* (<u>S</u>oci<u>a</u>l <u>N</u>etwork <u>Gree</u>dy <u>A</u>nonymization) algorithm, performs a greedy clustering processing to generate a *k*-anonymous masked social network, given an initial social network modeled as a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$.

Specifically, *SaNGreeA* puts together in clusters nodes that are as similar as possible in terms of their neighborhood structure. To do so, it uses a measure that quantifies the extent to which the neighborhoods of two nodes are similar with each other, i.e. the nodes manifest the same connectivity properties, or are connected / disconnected among them and with others in the same way.

To assess the proximity of two nodes' neighborhoods, we proceed as follows. Given $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, assume that nodes in $\mathcal{N}$ have a particular order, $\mathcal{N} = \{X_1, X_2, \ldots, X_n\}$. The neighborhood of each node X_i can be represented as an *n*-dimensional boolean vector $B_i = (b_1^i, b_2^i,, \ldots, b_n^i\ )$, where the j^{th} component of this vector, b_j^i, is 1 if there is an edge $(X_i, X_j) \in \mathcal{E}$, and 0 otherwise, $\forall j = 1, r; j \neq i$. We consider the value b_i^i to be *undefined*, and therefore not equal to 0 or 1.

The ***distance between two nodes*** (X_i and X_j) described by their associated *n*-dimensional boolean vectors B_i and B_j is:

$$dist(X_i, X_j) = \frac{|\{\ell | \ell = 1..n \wedge \ell \neq i,j;\ b_\ell^i \neq b_\ell^j\}|}{n-2}.$$

We exclude from the two vectors' comparison their elements *i* and *j*, which are undefined for X_i and respectively for X_j. As a result, the total number of elements compared is reduced by 2.

In the cluster formation process, our greedy approach will select the closest remaining node to be added to the cluster currently being formed. To assess the structural distance between a node and a cluster we use the below measure.

The ***distance between a node*** X ***and a cluster*** cl is defined as the average distance between X and every node from *cl*:

$$dist(X, cl) = \frac{\sum_{X_j \in cl} dist(X, X_j)}{|cl|}.$$

SaNGreeA algorithm creates one cluster at a time using the above measures. To form a new cluster, a node in $\mathcal{N}$ with the maximum degree and not yet allocated to any cluster is selected as a seed for the new cluster. Then the algorithm gathers nodes to this currently processed cluster until it reaches the desired cardinality *k*. At each step, the current cluster grows with one node. The selected node has to be unallocated yet to any cluster and it will minimize the distance between a node and a cluster measure.

It is possible, when *n* is not a multiple of *k*, that the last constructed cluster will contain less than *k* nodes. In that case, this cluster needs to be dispersed between the previously constructed groups. Each of its nodes will be added to the cluster that is closest to that node w.r.t. our previously defined distance measure.

The pseudocode of the *SaNGreeA* algorithm which modifies the algorithm from [7] by removing node attributes information is shown next.

```
Algorithm SaNGreeA is
 Input     $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ – a social network
           $k$ – as in $k$-anonymity
 Output    S = $\{cl_1, cl_2, \dots, cl_v\}$; $\bigcup_{j=1}^{v} cl_j = \mathcal{N}$ ; $cl_i \cap cl_j = \emptyset$,
           $i,j = 1..v$, $i \neq j$; $|cl_j| \geq k$, $j = 1..v$ - a set of
           clusters that ensures $k$-anonymity for $\mathcal{AG} = (\mathcal{AN}, \mathcal{AE})$
           so that a cost measure is optimized;

 S = $\emptyset$; $i$ = 1;
 Repeat
    $X^{seed}$ = a node with maximum degree from $\mathcal{N}$;
    $cl_i = \{X^{seed}\}$;
    // $\mathcal{N}$ keeps track of nodes not yet distributed to clusters
    $\mathcal{N} = \mathcal{N} - \{X^{seed}\}$;
    Repeat
       $X^* = \underset{X \in N}{\operatorname{argmin}}\, dist(X, cl_i)$;
       // $X^*$ is the node within $\mathcal{N}$ (unselected nodes) that
       // is closer to $cl_i$
       // $\mathcal{G}_1$ – the subgraph induced by $cl \cup \{X^*\}$ in $\mathcal{G}$;
       $cl_i = cl_i \cup \{X^*\}$;   $\mathcal{N} = \mathcal{N} - \{X^*\}$;
    Until ($cl_i$ has $k$ elements) or ($\mathcal{N} == \emptyset$);
    If ($|cl_i| < k$) then
       DisperseCluster(S, $cl_i$); // only for the last cluster
    Else
       S = S $\cup$ $\{cl_i\}$; $i$++;
```

```
      End If;
   Until $\mathcal{N} == \emptyset$;
   End SaNGreeA.

   Function DisperseCluster(S, $cl$)
      For every $X \in cl$ do
         $cl_u$ = FindBestCluster($X$, S); $cl_u = cl_u \cup \{X\}$;
      End For;
   End DisperseCluster;

   Function FindBestCluster($X$, S) is
      bestCluster = null; infoLoss = $\infty$;
      For every $cl_j \in$ S do
         If dist($X$, $cl_j$) < infoLoss then
            infoLoss = dist($X$, $cl_j$);
            bestCluster = $cl_j$;
         End If;
      End For;
      Return bestCluster;
    End FindBestCluster;
```

Figure 2 shows the anonymized network, $\mathcal{AG}_1$ that was obtained by applying SaNGreeA algorithm to the social network $\mathcal{G}_1$ (see Figure 1). Using this approach the nodes with the same color are clustered together in a supernode. This anonymized network satisfies 4-anonymity for social network property (k = 4).

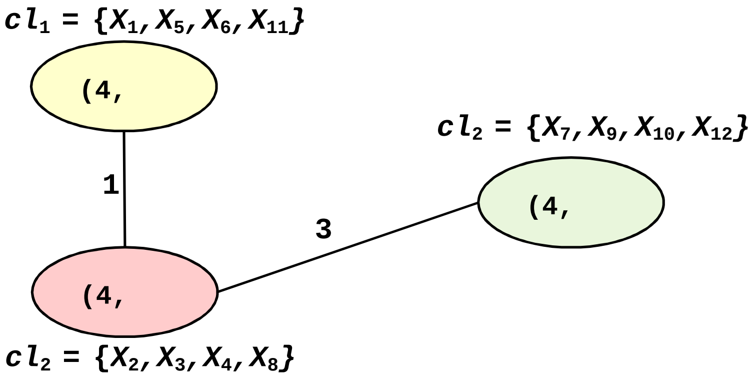
$cl_1 = \{X_1, X_5, X_6, X_{11}\}$
(4,
$cl_2 = \{X_7, X_9, X_{10}, X_{12}\}$
1
3
(4,
(4,
$cl_2 = \{X_2, X_3, X_4, X_8\}$

Figure 2. Anonymized social network, $\mathcal{AG}_1$.

3.2 K-Degree Anonymity

K-degree anonymity protects against intruders' attacks with background knowledge that is limited to nodes' degree. A social network is *k*-degree anonymous if for every node *X* in the network, there are at least *k*-1 other nodes with the same degree as the node *X* [18]. While an initial algorithm to create a *k*-degree anonymous network was proposed in [18], we used for this paper the *Fast K-Degree Anonymization* (*FKDA*) algorithm proposed by Lu et al. [19].

FKDA anonymizes a social network by adding edges in a greedy fashion until the network is *k*-degree anonymous. First, the nodes of the original graph are separated into several groups. Second, each predetermined group will be anonymized by adding edges to the nodes in the group until all the nodes in the group have the same degree. If anonymization cannot be achieved for a group in this edge creation algorithm, a more relaxed approach of adding edges is allowed, where nodes in the group being anonymized are connected to any nodes in the graph. The performing of the relaxed addition can destroy the anonymity of nodes processed in previous steps – and if this happens, the whole process is restarted from scratch. The time complexity for FKDA is $O(n^2)$ in the worst case, where *n* is the total number of nodes in the network. For complete details of the FKDA algorithm please consult [19].

Figure 3 illustrate the anonymized network, $\mathcal{AG}_2$ that was obtained by applying FKDA algorithm to the social network $\mathcal{G}_1$ (see Figure 1). The dashed red lines represent the new relationships added by FDKA algorithm. In this anonymized network the nodes X_1, X_2, X_3, and X_4 have degree 4; the nodes X_5, X_6, X_7, and X_8 have degree 2, and the remaining nodes X_9, X_{10}, X_{11}, and X_{12} have degree 1. This network satisfies 4-degree anonymity ($k = 4$).

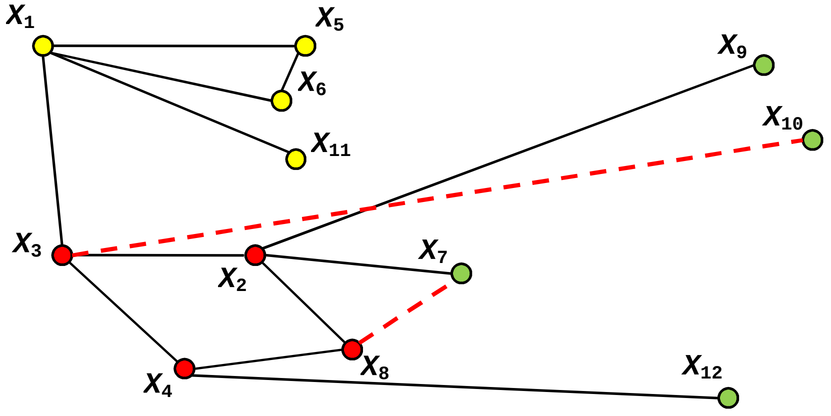

Figure 3. Anonymized social network, $\mathcal{AG}_2$.

It is worth noting that the privacy protection guaranteed by *k*-degree anonymization is "weaker" than the *k*-anonymity for social network model. The *k*-degree anonymization assumes that the only external structural information available to an intruder is the nodes' degree, while the *k*-anonymity for social network model assumes that complete structural information can be available (the entire graph structure without, of course, any sensitive information from the nodes' attributes). In other words, for *k*-anonymity for social network model, an intruder cannot identify which node from the corresponding supernode is the target individual, while this will not be true when more information will be available for *k*-degree anonymity model. This significant difference in privacy protection between the two models leads us to assume that *k*-degree anonymity model preserves better the communities than the *k*-anonymity for social network model. As we will see in the next Sections (and specifically in Section 6), this assumption is confirmed by our experiments.

4 De-anonymization Process

To compare communities between social networks and *k*-degree anonymous social network is easier since both the initial and anonymized networks have the same number of nodes and only the number of edges differ (see Figure 1 and Figure 3). This comparison is more difficult in case of *k*-anonymous social networks because the number of nodes in the anonymized network is reduced by a factor of *k* from the initial social network. To avoid this problem we "de-anonymize" *k*-anonymous social networks using two different models by trying to reverse the anonymization process and create replicas of the original network. The de-anonymized networks will have the same number of nodes and edges as the original network, allowing therefore for a fair comparison of communities.

Two possible de-anonymized social networks of the anonymized network $\mathcal{AG}_1$ (see Figure 2), labeled $\mathcal{DG}_1$ and $\mathcal{DG}_2$, are shown in Figures 4 and 5. Notice that they have the same number of nodes and edges as the initial social network $\mathcal{G}_1$, but they have a different structure.

To de-anonymize a *k*-anonymous social network we re-use the two methods presented in [1, 32], *Uniform De-anonymization* [32] and *R-MAT De-anonymization* [1].

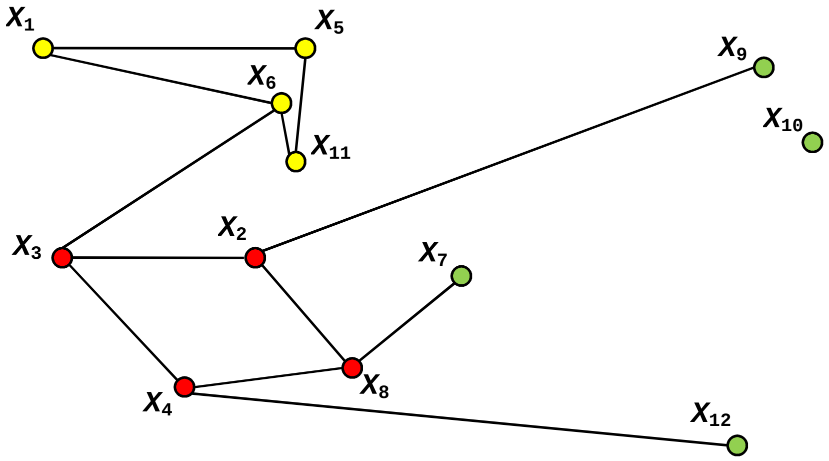

Figure 4. De-anonymized social network, $\mathcal{DG}_1$.

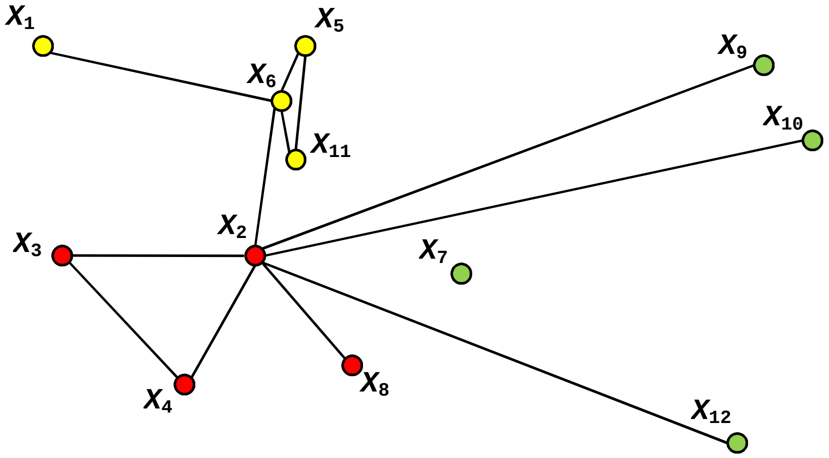

Figure 5. De-anonymized social network, $\mathcal{DG}_2$.

Uniform De-anonymization will randomly create edges between nodes within each super-node up to the number of edges in that super-node, and between nodes from different super-nodes until the number of generated edges corresponds with the super-edge weight (similar with Erdos-Renyi random graph generator method).

The R-MAT De-anonymization method is based on the assumption that many real-world networks are scale-free, and their nodes degree distribution follows a power-law. R-MAT De-anonymization method takes an empty adjacency matrix of a graph and recursively determines the location of a new edge in this matrix. To select the placement of e new edge, the algorithm divides the adjacency matrix into 4 equal-sized partitions and the location of the new edge is probabilistically selected in one of the 4 locations, based on four probability parameters, called *a*, *b*, *c*, and *d*. Once a partition is selected, it is again divided into 4 sub-partitions until we reach a simple cell (=1×1 partition). R-MAT De-anonymization works on a submatrix of the adjacency matrix of $\mathcal{G}'$ which is: a restriction of it to a cluster (to generate internal edges in that cluster), or a restriction of it to two clusters (to generate inter-cluster edges).

If an edge was already placed on the newly selected edge location, the procedure will restart from the beginning (since multiple edges between the same pair of nodes are not allowed in our graph model). For all our tests we used the following values for the four probability parameters: 0.45, 0.15, 0.15, and 0.25. This choice seems to model better many real-world graphs that follow power-law degree distributions [9]. As explained in [9], this generation technique will create two large well-connected "communities" in the graph: one among the nodes in the first "half" of the node set (the top-left quadrant in the adjacency matrix), the other among the nodes in the second half of the node set (the bottom-right quadrant in the adjacency matrix). Edges are created with higher probability among nodes in those respective halves, since parameters *a* and *d* are higher. The two communities are more loosely connected, as decided by the lower probabilities *b* and *c* that command the placement of edges between nodes belonging to different halves. The process is repeated recursively in each quadrant such that larger communities are divided in smaller and smaller communities.

5 Community Detection

In this paper we study how well anonymized social network preserve existing communities from the original social networks. We chose to focus on a specific community detection method known as Louvain method [4, 27] which is a heuristic algorithm based on modularity optimization [23]. This community detection method is implemented in the social network analysis software, Pajek, which we used for our experiments [25]. The modularity is a quality function that can be computed for a graph partitioned in communities. This modularity function is defined for a social network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ as follows [23]:

$$Q = \frac{1}{2m} \sum_{\substack{i,j=1 \\ i \neq j}}^{n} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j),$$

where

- n represents the number of nodes ($n = |\mathcal{G}|$);
- m represents the number of edges ($m = |\mathcal{E}|$);
- c_i and c_j represents the communities to which nodes X_i and X_j have been assigned;
- A_{ij} represents whether there is an edge between nodes X_i and X_j ($A_{ij} \neq 0$) or not ($A_{ij} = 0$);
- k_i and k_j represents the degree of nodes X_i and X_j;
- $\delta(c_i, c_j)$ is 1 if nodes X_i and X_j belong to the same community (c_i = uc_j) and 0 otherwise.

Since the terms from the modularity sum are non-zero only for nodes from the same community, the modularity function can be rewritten as [12]:

$$Q = \sum_{c=1}^{n_c} \left[\frac{l_c}{m} - \left(\frac{d_c}{2m} \right)^2 \right],$$

where

- n_c represents the number of communities;
- l_c represents the total number of edges joining nodes from community c (inter-cluster edges);
- d_c represents the sum of the degrees of nodes from c.

As stated in [12], $\frac{l_c}{m}$ is the actual fraction of edges in the network inside the community and $\left(\frac{d_c}{2m}\right)^2$ is the expected fraction of edges that would be there if the network will be a random network with same expected degree for each node.

This modularity function has a drawback that sometimes creates communities that contains very dense communities that are weakly connected [12]. In such case it might be more appropriate to consider the dense communities as individual communities. To alleviate this problem, a resolution parameter r was introduced and the new modularity function is defined as [12]:

$$Q_r = \sum_{c=1}^{n_c} \left[\frac{l_c}{m} - r\left(\frac{d_c}{2m}\right)^2 \right].$$

When resolution parameter is greater than 1 then larger number of smaller communities is desired, when resolution parameter is less than 1 then smaller number of larger communities is sought. Of course, the value 1 is equivalent with the original definition of modularity function.

A modularity-based community detection algorithm will try to find a set of communities that will maximize the modularity function. Unfortunately, the optimal solution is an NP-complete problem [6], and existing algorithms are based on heuristic solutions such as greedy techniques, simulated annealing, extremal optimization, and spectral optimization [12].

In this paper we use a heuristic method based on modularity optimization known as Louvain implementation [4, 27] from Pajek 3.14 [25]. While this implementation allows changing the resolution parameter, we chose to use only the default value, 1, in other words we used the original modularity function as the optimization criterion. This algorithm is divided into two phases that are repeated iteratively. In the first phase each node is assigned to one community and then nodes are moved between communities in such a way that the modularity gain is maximized. After a series of moves no node move will create a modularity gain. In the second phase, a weighted network is built from the network obtained at the end of the first phase. In this weighted network, one node represents a community from the original network, and weights are added to edges to represent the number of original edges that are collapsed into a super-edge. Once this phase is completed, then the first phase of the algorithm will be reapplied to this new network. The process of repeating these two phases will stop when the modularity is maximized. More detailed about this algorithm as well as an example can be found in [4].

5.1 Community Preservation

Using Louvain method we can compute communities for the initial social networks, the *k*-degree anonymous social networks (Section 3.2), and the de-anonymized *k*-anonymous social networks (Sections 3.1 and 4). To compare the results between an anonymized social network and the corresponding initial social network we use two different approaches. In the first approach named *naive community preservation*, we simply count how many nodes from the original communities remained in the same community after the processes of anonymization and de-anonymization. We illustrate this approach with the following example. Figure 6 shows the initial social network, labeled $\mathcal{SN}_1$. Figure 7 shows a social network that was obtained from the initial social network by applying SaNGreeA algorithm with k = 2 and then the R-MAT De-anonymization procedure ($\mathcal{SN}_2$). Figure 8 shows a 2-degree anonymous social network obtained by applying FKDA algorithm with k = 2 ($\mathcal{SN}_3$). For simplicity, we represent the nodes with number labels. In these figures, the communities are illustrated by different node colors.

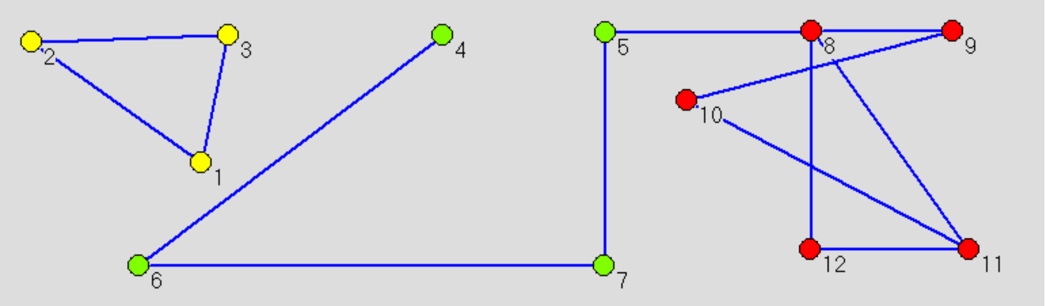

Figure 6. Initial social network, $\mathcal{SN}_1$.

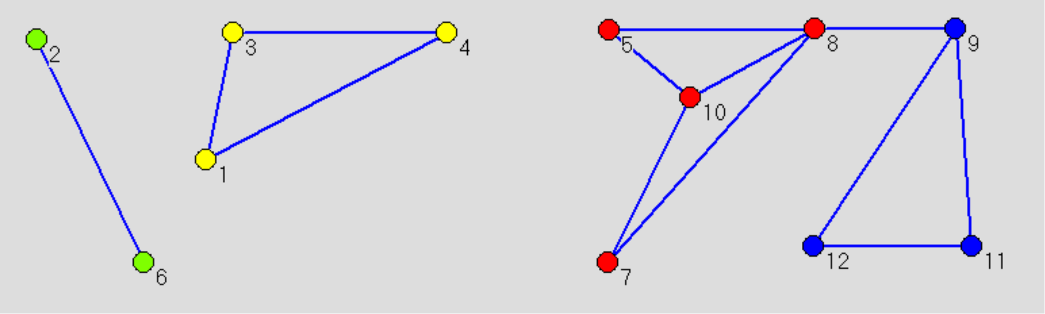

Figure 7. De-anonymized social network, $\mathcal{SN}_2$.

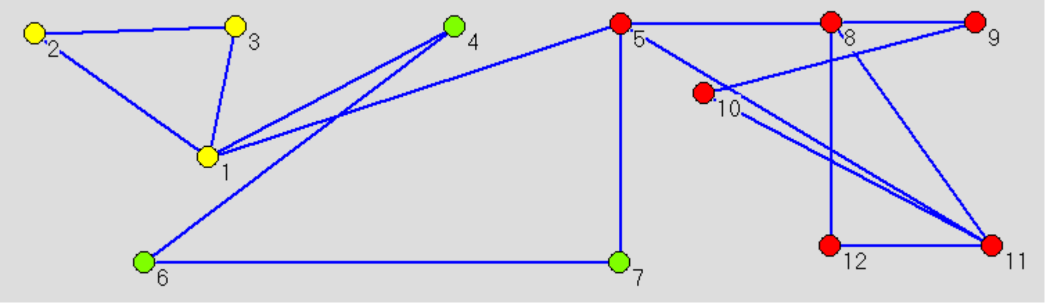

Figure 8. 2-degree anonymous social network, $\mathcal{SN}_3$.

Table 1 shows the communities and how they are preserved between $\mathcal{SN}_1$ and $\mathcal{SN}_2$, in other words for *k*-anonymity for social networks privacy model. Table 2 illustrate the communities and how they are preserved between $\mathcal{SN}_1$ and $\mathcal{SN}_3$, in other words for *k*-degree anonymity privacy model. The communities were obtained using Louvain method.

To compute the % preservation column, for each community from $\mathcal{SN}_1$ we select a corresponding community from $\mathcal{SN}_2$ or $\mathcal{SN}_3$ that contain the maximum number of elements from the initial community. For instance for the third community from Table 1, {8, 9, 10, 11, 12}, the best match is the community {9, 11, 12} and the % preservation is 3/5. To find out the naive community preservation measure (labeled *NCP*) we average the results from the % preservation column and we obtain the following results:

- $NCP(\mathcal{SN}_1, \mathcal{SN}_2) = 58.66\%$
- $NCP(\mathcal{SN}_1, \mathcal{SN}_3) = 91.66\%$.

Table 1. Naive community preservation – *k*-anonymity for social networks

Community ID	Communities in $\mathcal{SN}_1$	Communities in $\mathcal{SN}_2$	% Preservation
1	1, 2, 3	1, 3, 4	66%
2	4, 5, 6, 7	5, 7, 8, 10	50%
3	8, 9, 10, 11, 12	9, 11, 12	60%

Table 2. Naive community preservation – *k*-degree anonymity

Community ID	Communities in $\mathcal{SN}_1$	Communities in $\mathcal{SN}_2$	% Preservation
1	1, 2, 3	1, 2, 3	100%

2	4, 5, 6, 7	4, 6, 7	75%
3	8, 9, 10, 11, 12	5, 8, 9, 10, 11, 12	100%

In the second approach, we consider the community preservation for each node individually. For any node, we compare the original and final communities to which the node belongs. We call this second measure *community preservation at node level* and we label it *CPNL*. We formalize this approach as follows. We represent the original social network as $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ and the final social network as $\mathcal{G}' = (\mathcal{N}, \mathcal{E}')$. Note that the nodes from the initial and final social networks are identical for both anonymization models presented in this paper (this is not exactly true for *k*-anonymity for social network model, but the de-anonymization process presented in Section 4 recreates the same number of nodes). The set of nodes $\mathcal{N}$ is represented as = $\{X_1, X_2, .., X_n\}$. For each node X_i, we represent the communities in which the node belongs as $C(X_i)$ and $C'(X_i)$. The community preservation for the node X_i is defined as the number of nodes that are common between $C(X_i)$ and $C'(X_i)$ divided to all the nodes that belong to at least one of these two communities. In other words,

$$CPNL(X_i) = \frac{|C(X_i) \cap C'(X_i)|}{|C(X_i) \cup C'(X_i)|},$$

where |S| represents the cardinality of the set S.

The final community preservation at node level measure is computed as the average of all individual values as shown below:

$$CPNL(G, G') = \frac{\sum_{i=1}^{n} CPNL(X_i)}{n},$$

We illustrate this approach using the example from Figures 6 – 8. Table 3 shows for each node the original and final communities between $\mathcal{SN}_1$ and $\mathcal{SN}_2$ (for *k*-anonymity for social networks privacy model) and the corresponding community preservation at node level value. Table 4 shows for each node the original and final communities between $\mathcal{SN}_1$ and $\mathcal{SN}_3$ (for *k*-degree anonymity privacy model) and the corresponding community preservation at node level value. To illustrate this computation let us consider the node with ID 5 from Table 3 (comparison between $\mathcal{SN}_1$ and $\mathcal{SN}_2$). The original community for this node is {4, 5, 6, 7} and the final community is {5, 7, 8, 10}. There are two nodes in the intersection of those two sets ({5, 7}) and six nodes in their union ({4, 5, 6, 7, 8, 10}). Thus, *CPNL*(Node ID = 5) = 2/6. To find out the final community preservation at node level measure (*CPNL*) we average the results from the $CPNL(X_i)$ column and we obtain the following results:

- $CPNL(\mathcal{SN}_1, \mathcal{SN}_2) = 38.79\%$
- $CPNL(\mathcal{SN}_1, \mathcal{SN}_3) = 79.40\%$

Table 3. Community preservation at node level – *k*-anonymity for social networks

Node ID	$C(X_i)$	$C'(X_i)$	$CPNL(X_i)$
1	1, 2, 3	1, 3, 4	50%
2	1, 2, 3	2, 6	25%
3	1, 2, 3	1, 3, 4	50%
4	4, 5, 6, 7	1, 3, 4	16.66%
5	4, 5, 6, 7	5, 7, 8, 10	33.33%
6	4, 5, 6, 7	2, 6	20%
7	4, 5, 6, 7	5, 7, 8, 10	33.33%
8	8, 9, 10, 11, 12	5, 7, 8, 10	28.57%
9	8, 9, 10, 11, 12	9, 11, 12	60%
10	8, 9, 10, 11, 12	5, 7, 8, 10	28.57%
11	8, 9, 10, 11, 12	9, 11, 12	60%
12	8, 9, 10, 11, 12	9, 11, 12	60%

Table 4. Community preservation at node level – *k*-degree anonymity

Node ID	$C(X_i)$	$C'(X_i)$	$CPNL(X_i)$
1	1, 2, 3	1, 2, 3	100%
2	1, 2, 3	1, 2, 3	100%
3	1, 2, 3	1, 2, 3	100%
4	4, 5, 6, 7	4, 6, 7	75%
5	4, 5, 6, 7	5, 8, 9, 10, 11, 12	11.11%
6	4, 5, 6, 7	4, 6, 7	75%
7	4, 5, 6, 7	4, 6, 7	75%
8	8, 9, 10, 11, 12	5, 8, 9, 10, 11, 12	83.33%
9	8, 9, 10, 11, 12	5, 8, 9, 10, 11, 12	83.33%

10	8, 9, 10, 11, 12	5, 8, 9, 10, 11, 12	83.33%
11	8, 9, 10, 11, 12	5, 8, 9, 10, 11, 12	83.33%
12	8, 9, 10, 11, 12	5, 8, 9, 10, 11, 12	83.33%

6 Experiments and Results

We study the preservation of communities between original and anonymized/de-anonymized versions of the following publically available datasets:

- **Cond** is a collaboration network of scientists [22]. This network is undirected and consists of 16,726 nodes, 47,594 edges, and 1247 communities. The number of communities is obtained using Louvain method from Pajek network analysis tool. Two scientists are considered connected (have an edge between them) if they coauthored a paper.
- **Enron** dataset is a network of email exchanges [15, 16]. It is an undirected network with 36,692 nodes, 183,831 edges, and 1286 communities. Each node in this network represents an email address. An edge exists between two nodes if at least one email was sent from one node to the other from that edge.
- **YouTube** dataset is an undirected social network [36]. The network has 1,157,827 nodes and 2,987,624 edges. Due to the large number of nodes and edges in the network, we extracted three sub-graphs from it. Each sub-graph is a well-defined community from the original network. Again, we used Louvain method from Pajek to extract the communities. YouTube network has 30,814 communities. Only six of these communities have number of nodes in the range between 15,000 and 40,000 which is the range of nodes we look for in our experiments. We will refer to these communities as the preferred-communities. When creating a sub-graph for a community, we retained only the nodes that members of the specified community and the edges that connect these selected nodes.

After creating the sub-graphs for the preferred-communities, we chose three sub-graphs as our initial social networks based on a unique feature for each one of them. Following is the description of these networks:

- **YouTubeLargest** is the largest community in YouTube preferred-communities. It has 37,530 nodes, 121,337 edges, and 363 communities. We used the number of nodes to measure the size of the communities and determine the largest one.

- **YouTubeCompact** is the most compact community from YouTube preferred-communities. We used the Clustering Coefficient to measure the compactness of the network. When using Pajek to measure the Clustering Coefficient [33, 34] for YouTubeCompact, Watts-Strogatz Clustering Coefficient was 0.24883441 and Network Clustering Coefficient (Transitivity) was 0.04206904, which are the largest values among the other communities in the preferred-communities. YouTubeCompact contains 20,272 nodes, 28,026 edges, and 128 communities.
- **YouTubeRandom** is a community that was chosen randomly from YouTube network preferred-communities. It has 22,409 nodes, 27,927 edges and 143 communities.

The workflow of our experiments is shown in Figure 9. We describe next all the steps in this workflow that are performed in order to measure the community preservation.

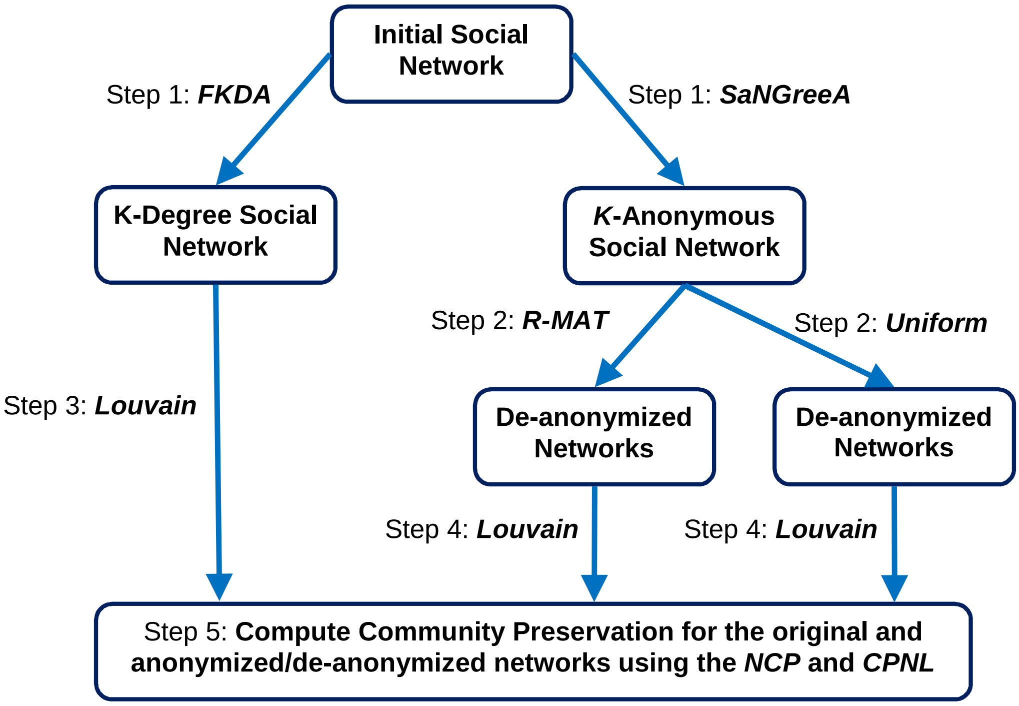

Figure 9. Workflow for community preservation experiments.

Step 1: First, we started with the initial networks (**Cond**, **Enron**, **YouTubeLargest**, **YouTubeCompact**, **YouTubeRandom**) described previously. We anonymized these networks with FKDA and SaNGreeA using several anonymity parameter *k:* 5, 10, 15, 20, 25, and 50.

Step 2: For each *k*-anonymous social network we generated 5 de-anonymized networks using Uniform De-anonymization and 5 de-anonymized networks using R-MAT De-anonymization (Section 4). Repeating the de-anonymization process 5 times was done because of the randomness of the de-anonymization process. In this step, we also run the de-anonymization processes for a *k*-anonymous social network with $k = n$ (size of the network), this is equivalent with executing Uniform and R-MAT De-anonymization without having any knowledge regarding the initial network structure except its size (the number of nodes and the number of edges).

Step 3: After that, we extracted the communities of the original networks using Louvain community detection method in Pajek using the following steps: Network-> create partition->Communities->Louvain Method-> Multi- Level Coarsening + Multi- Level Refinement.

Step 4: Then, we extracted the communities from *k*-degree anonymous networks and the de-anonymized networks as described in the previous step.

Step 5: To compute the community preservation, we used two different approaches:

- In the first approach, *naive community preservation (NCP),* we mapped every community detected in the original network to the best match community in the anonymized/de-anonymized networks. A best match community would be a community that has the most nodes from the original community. After that, we compute the percentage of nodes that remain the same community before and after the anonymization/de-anonymization process. Finally, we take the average community preservation for all the communities in the original network.
- The second approach we measure the community preservation at the node level (*CPNL*). For a node *X*, we divide the number of nodes in common between the original and the anonymized community for node *X* by the number of nodes that belong to either the original or the anonymized community of node *X* or to both. We do this computation for all the nodes in the graph. The final result for the *CPNL* is averaged.

An example for the process of each one of these two approaches is shown in Section 5. Since we generated 5 de-anonymized networks for each *k*-anonymous social network, the community preservation determined in those cases is averaged.

To execute our experiments, we implemented SaNGreeA (used in Step 1), FKDA (Step 1), Uniform and R-MAT De-anonymization (Step 2), and computation of the NCP and CPNL measures (Step 5) in Java. The Louvain community detection algorithm (Step 3 and 4) was executed using Pajek. All the experiments were performed on an Intel® Core™ i7-2640M 2.80 GHz CPU machine with 8 GB memory running on 64 bit Windows 7 Home Premium.

The average community preservation (% preservation) results for the *NCP* experiments are shown in Figures 10-14, left side, for Cond, Enron, YouTubeLargest, YouTubeCompact, and YouTubeRandom datasets. The vertical axis represents the percentage of the average community preservation using *NCP* measure for the networks. The last *k* value represents the size of the network and we report in this case the community preservation when there is no *k*-anonymous social network available; in other words all the nodes and edges are collapsed into a super-node where the number of nodes and the number of edges for the entire initial network are reported. The community preservation for this case represents the baseline value, and in all experiments the community preservation is superior to this baseline case.

Figures 10-14, right side, show the improvement factor for *k*-anonymity for social network model when de-anonymization is performed using both R-MAT and Uniform approaches compared with the communities that exist in a random graph (uniform random graph and R-MAT random graph) with the same number of node and vertices (the improvement factor for the baseline case is 1) using the *NCP* measure. Such an improvement factor computation cannot be done for *k*-degree anonymization model since there is no graph generation performed in that case.

For Cond network (Figure 10), FKDA had a good preservation of the communities of the original network and there were a noticeable decrease only in the case were *k* = 50. On the other hand, R-MAT and Uniform de-anonymization had almost identical preservation for the communities of the original network except for the case where *k* = 5, R-MAT had much better preservation than Uniform.

For Enron network (Figure 11), FKDA preserved the communities of the original network very well. R-MAT De-anonymization preserved the communities of the original networks well when *k* was small and the community preservation started to drop rapidly as *k* got larger. Uniform De-anonymization had the lowest preservation of communities when *k* was 5 and 10, but for the larger values of *k*, Uniform performed slightly better than R-MAT.

FKDA also preserved the communities well for YouTubeLargest network for all *k* values (Figure 12). And as with Cond network, R-MAT performed better when *k* was 5 but for the larger values of *k* R-MAT and Uniform had almost the same preservation.

For YouTubeCompact (Figure 13) and YouTubeRandom (Figure 14) we had similar curves for FKDA, R-MAT De-anonymization, and Uniform De-anonymization. FKDA had the best preservation of communities followed by R-MAT De-anonymization. For both of these cases the preservation of communities decreased continuously. However, Uniform De-anonymization had the worst community preservation with an almost steady line.

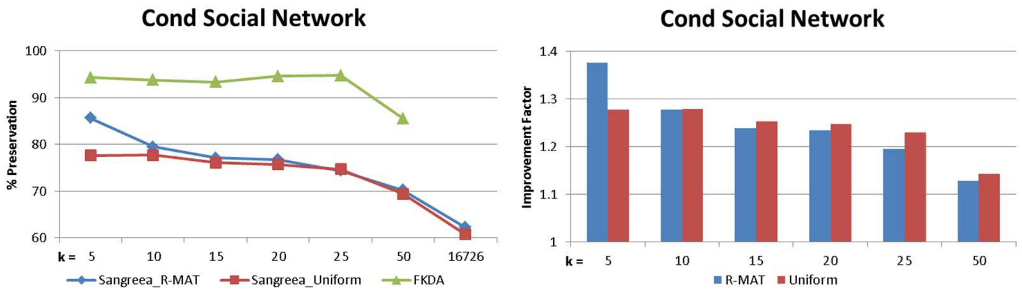

Figure 10. % NCP preservation and improvement factor for Cond.

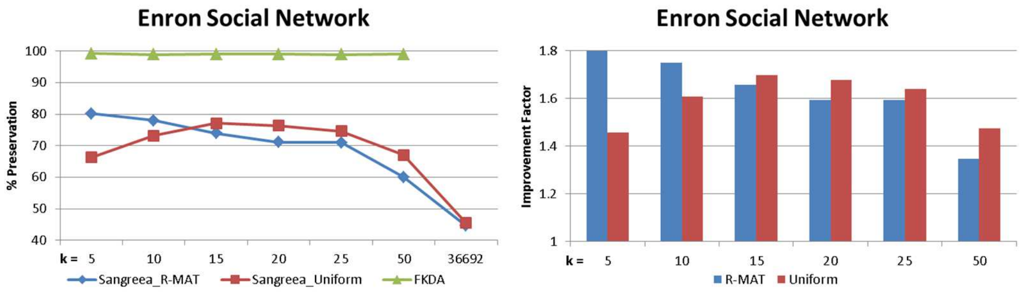

Figure 11. % NCP preservation and improvement factor for Enron.

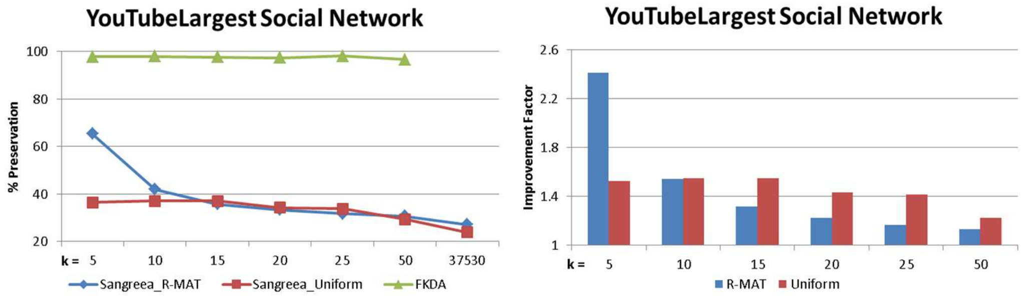

Figure 12. % NCP preservation and improvement factor for YouTubeLargest.

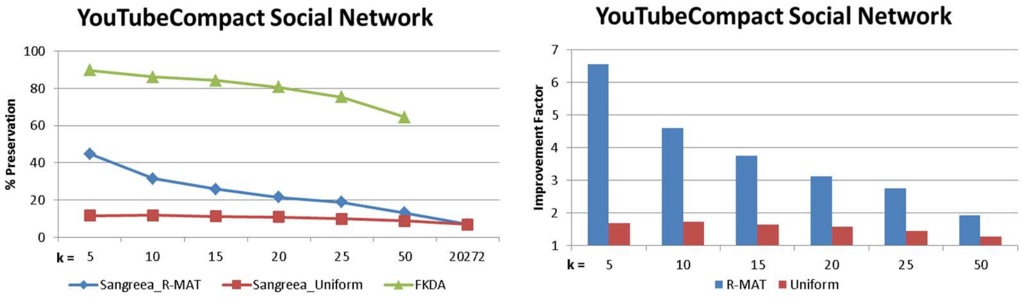

Figure 13. % NCP preservation and improvement factor for YouTubeCompact.

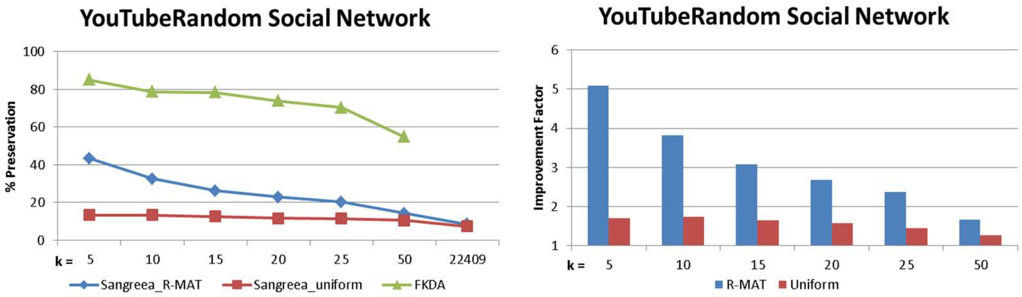

Figure 14. % NCP preservation and improvement factor for YouTubeRandom.

Both Uniform and R-MAT methods had a good improvement factor compared to the baseline (Figures 10-14, right side). R-MAT is outperforming Uniform for small values of *k*. With the exception of YouTubeCompact and YouTubeRandom, Uniform De-anonymization had better improvement factor than R-MAT for the larger *k* values (*k*=15 to *k*=50).

Figures 15-19, left side, show the results for the *CPNL* measure for Cond, Enron, YouTubeLargest, YouTubeCompact, and YouTubeRandom datasets. As with the previous measure, the vertical axis represent the percentage of *CPNL* and the horizontal axis represent the *k* value. Figures 15-19, right side, show the improvement factor for R-MAT and Uniform using CPNL measure.

Figure 15 shows the result for *CPNL* with Cond network. R-MAT had the best *CPNL* for *k* =5 and *k* =10 and it starts to decrease linearly as *k* grew larger. FKDA had much worse *CPNL* than the *NCP* for the same network. However, Uniform had the worst *CPNL*. It is very weak in preserving the communities at the node level.

For Enron network (Figure 16) FKDA had the best *CPNL* followed by R-MAT. As with Cond network, Uniform is not preserving the communities at the node level very well.

Figure 17 shows the *CPNL* for YouTubeLargest network. FKDA had a good *CPNL* with the best result when *k*=15. R-MAT preservation for the communities at the node level decreased as *k* grew larger and it performed better that Uniform where the latter had an almost straight line.

Similarly for the *NCP*, *CPNL* for YouTubeCompact (Figure 18) and YouTubeRandom (Figure 19) had similar curves for FKDA, R-MAT, and Uniform. FKDA had the best performance with almost linear line whereas R-MAT and Uniform are nearly the same line with R-MAT performing slightly better than Uniform for *k*=5 to *k*=25.

Based on the improvement factor computation (Figures 15-19, right side), R-MAT showed significant improvement compared to the baseline. On the other hand, Uniform did not show good improvement with the exception of YouTubeLargest network.

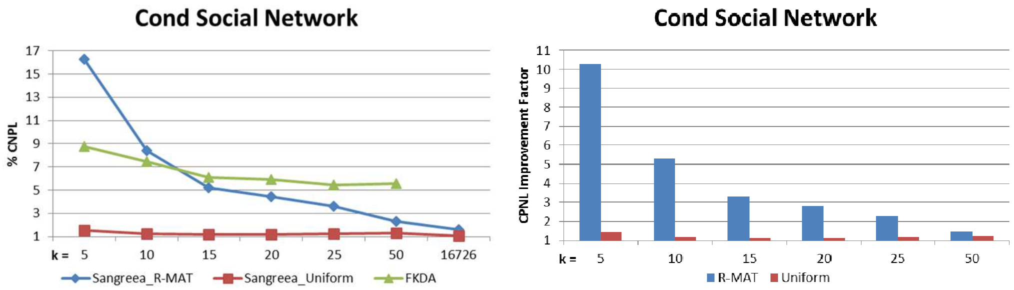

Figure 15. % CPNL and improvement factor for Cond.

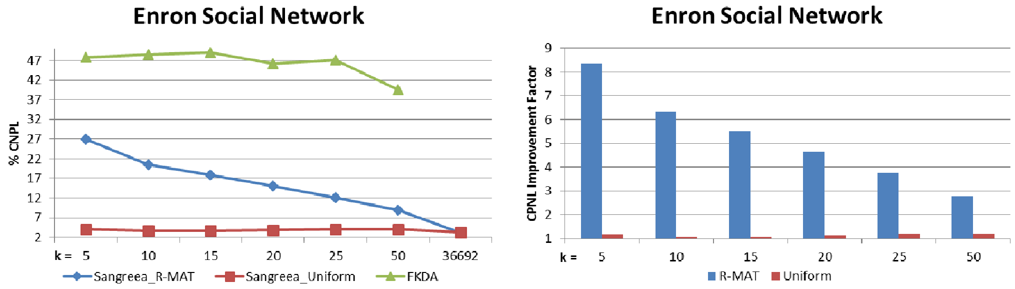

Figure 16. % CPNL and improvement factor for Enron.

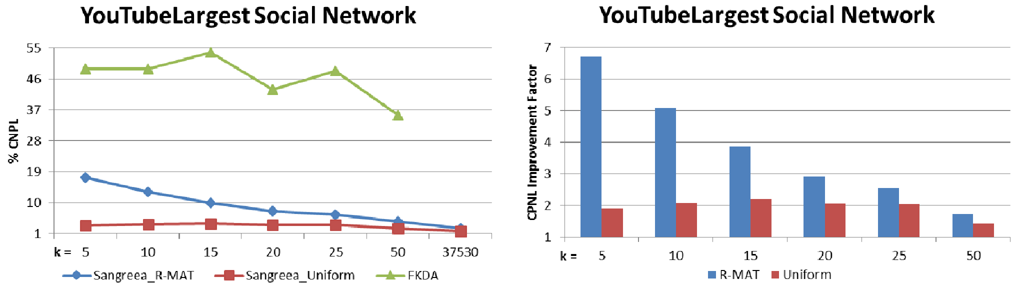

Figure 17. % CPNL and improvement factor for YouTubeLargest.

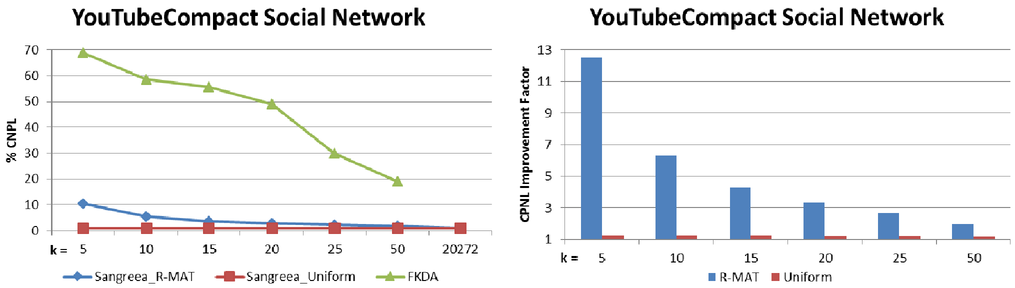

Figure 18. % CPNL and improvement factor for YouTubeCompact.

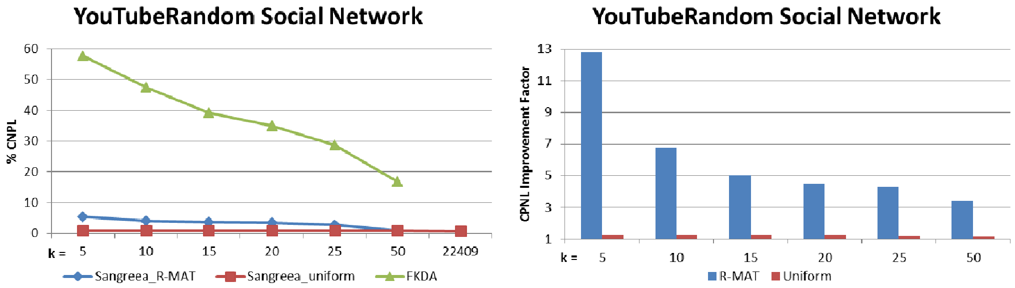

Figure 19. % CPNL and improvement factor for YouTubeRandom.

Based on the results reported in Figures 10-19, we conclude that FKDA algorithm preserves very well the community structure of the initial social network using both *NCP* and *CPNL* measures. This result is expected since *k*-degree anonymity keeps most of the initial structure of the social network. However, as pointed out in Section 3, *k*-degree anonymity is a "weak" anonymity model since it assumes that an intruder has only knowledge about the degree of individuals in the network and not about the network structure. The other two methods used in conjunction with *k*-anonymity for social network model (Uniform and R-MAT De-anonymization) while clearly outperformed by FKDA, also preserves to some extent the community structure of the original network. As expected R-MAT De-anonymization is, in general, outperforming Uniform De-anonymization.

As expected, the smaller the value of *k*, the communities are better preserved using both *NCP* and *CPNL*. However, this is not true for some of the experiments. For FKDA, since the results are very similar for all values of *k*, in some cases the % preservation increases when *k* increases. This is due to addition of edges within original communities for larger *k* which contribute to their preservation in the anonymized dataset. For de-anonymization the only such inversion is detected for Enron dataset and Uniform De-anonymization method using *NCP* measure. This is likely because the SaNGreeA algorithm breaks larger communities in super-nodes of size *k*, and then the Uniform De-anonymization will generate edges between vertices from different communities such that the initial communities cannot be found in the final de-anonymized networks. R-MAT De-anonymization is able to better preserve such community due to its edge generation procedure that follows better the degree distribution of the initial network.

It is also worth noting that in all three experiments that use YouTube dataset, the communities are well preserved in case of R-MAT De-anonymization and low *k* values, in particular for YouTubeCompact and YouTubeRandom, the improvement factor is over 5 (for $k = 5$) using *NCP* and over 6 for all the networks using *CPNL* (also for k=5). This is due to a combination of factors. First, as stated above, the R-MAT De-anonymization is preserving the original network structure better. And second, the communities are not well preserved in case of a random graph, thus the % preservation is very low for the baseline case.

7 Conclusions and Future Work

In this paper, we studied how well communities are preserved when social networks are anonymized. We analyzed two models *k*-anonymity for social networks and *k*-degree anonymity. Our results show that FKDA algorithm used to create a *k*-degree anonymous network preserved very well the communities from the initial networks. The de-anonymization methods used after the social networks were anonymized with SaNGreeA algorithm (to became *k*-anonymous social networks) also are able to preserve, although less successfully than FKDA, the initial communities. In most experiments the R-MAT de-anonymization outperforms the Uniform De-anonymization.

From the privacy point of view, *k*-anonymity for social networks enforces a much stronger model than *k*-degree anonymity. *K*-degree anonymity only considers the degree of each node as possible background knowledge for an intruder; so an intruder with more knowledge about the network structure can breach the privacy of a *k*-degree anonymous network. For *k*-anonymous networks, an intruder with any background knowledge about the structure of the network cannot breach the privacy of the network.

There are several future research directions that we want to pursue. First, the community preservation measure is useful when the number of communities is roughly the same between the initial and anonymized social network. When the number of communities in the anonymized social network decreases it is likely that the original communities are preserved in larger communities. Our measure does not distinguish between these two situations and, therefore, we intend to create a more robust way of comparing communities' preservation. Second, the criterion to construct super-nodes in SaNGreeA is based on neighbor similarities between all nodes from the network. We intend to adapt SaNGreeA algorithm to create super-nodes with nodes that belong to one community, and in this way we hope to increase the community preservation.

References

[1] Alufaisan, Y., Campan, A. (2013) Preservation of Centrality Measures in Anonymized Social Networks, Proceedings of the ASE/IEEE International Conference on Privacy, Security, Risk, and Trust (PASSAT 2013), Washington D.C., USA.

[2] Arenas, A., Duch, J., Fernandez, A., Gomez, S. (2007) Size Reduction of Complex Networks Preserving Modularity, New J. Phys. 9, art. no. 176.

[3] Bhagat, S., Cormode, G., Krishnamurthy, B., Srivastava, D. (2009) Class-based Graph Anonymization for Social Network Data, Proceedings of the International Conference on Very Large Data Bases (VLDB).

[4] Blondel, V. D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E. (2008) Fast Unfolding of Communities in Large Networks, Journal of Statistical Mechanics: Theory and Experiment, 1742-5468.

[5] Bollobás, B. (2001) Random Graphs, 2nd ed., Cambridge University Press.

[6] Brandes, U., Delling, D., Gaertler, M., Gorke, R., Hoefer, M., Nikoloski, Z., Wagner, D. (2008) On Modularity Clustering, IEEE Transactions on Knowledge and Data Engineering, Vol. 20, No. 2, February 2008, 172-188.

[7] Campan, A., Truta, T. M. (2008) A Clustering Approach for Data and Structural Anonymity in Social Networks, Proceedings of the 2nd ACM SIGKDD International Workshop on Privacy, Security, and Trust in KDD (PinKDD).

[8] Campan, A., Alufaisan, Y. (2013) Social Network Anonymization and Influence Preservation, Proceedings of the International Conference on Data Mining (DMIN'13), Las Vegas, Nevada, USA.

[9] Chakrabarti, D., Zhan, Y., Faloutsos, C. (2004) R-MAT: A Recursive Model for Graph Mining, Proceedings of the SIAM International Conference on Data Mining (SDM'04), 442-446.

[10] Chakrabarti, D., Faloutsos, C. (2006) Graph Mining: Laws, Generators, and Algorithms, ACM Computing Surveys, Volume 38, Article 2.

[11] Cheng, J., Fu, A. W. C., Liu, J. (2010) K-isomorphism: Privacy Preserving Network Publication against Structural Attacks, Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, 459-470, DOI= http://doi.acm.org/10.1145/1807167.1807218.

[12] Fortunato, S. (2010) Community Detection in Graphs, Physics Reports, Volume 486, Issues 3–5, 75-174, DOI=http://dx.doi.org/10.1016/j.physrep.2009.11.002.

[13] Freeman, L. C. (1979) Centrality in Social Networks: Conceptual Clarification, Social Networks, vol. 1, no. 3, 215-239.

[14] Harary, F. (1994) Graph Theory, Addison-Wesley.

[15] Klimmt, B., Yang, Y. (2004) Introducing the Enron Corpus, CEAS conference.

[16] Leskovec, J., Lang, K., Dasgupta, A., Mahoney, M. (2009) Community Structure in Large Networks: Natural Cluster Sizes and the Absence of Large Well-defined Clusters, Internet Mathematics, Vol. 6, No 1, 29-123.

[17] Leskovec, J., Lang, K.L, Mahoney, M.W. (2010) Empirical Comparison of Algorithms for Network Community Detection, Proceedings of the World Wide Web Conference (WWW 2010), Raleigh, North Carolina USA, 631-640.

[18] Liu, K., Terzi, E. (2008) Towards Identity Anonymization on Graphs, Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, 93-106, DOI= http://doi.acm.org/10.1145/1376616.1376629.

[19] Lu, X., Song, Y., Bressan, S. (2012) Fast Identity Anonymization on Graphs, Proceedings of the 23rd International Conference on Database and Expert Systems Applications (DEXA), 281-295.

[20] Lukovits, I., Nikolic, S., Trinajstic, N. (2002) On Relationships between Vertex-Degrees, Path-Numbers and Graph Valence-Shells in Trees, Chemical Physics Letter, Vol. 354, 417-422.

[21] Massen, C. P., Doye, J. P. K. (2006) Thermodynamics of Community Structure, ePrint arXiv:cond-mat/0610077.

[22] Newman, M. E. J. (2001) The Structure of Scientific Collaboration Networks, Proc. Natl. Acad. Sci. USA 98, 404-409.

[23] Newman, M. E. J., Girvan, M. (2004) Finding and Evaluating Community Structure in Networks, Physical Review, E 69 (2), 026113.

[24] Noack, A. (2009) Modularity Clustering is Force-Directed Layout, Physical Review, E 79 (2), 026102.

[25] Nooy, W., Mrvar, A., Batagelj, V. (2011) Exploratory Social Network Analysis with Pajek, Revised and Expanded Second Edition, Structural Analysis in the Social Sciences, Vol. 34, Cambridge University Press, 2011.

[26] Olson, P. (2013) Teenagers Say Goodbye to Facebook and Hello to Messenger Apps, The Observer Journal, Saturday 9 November 2013, Online at: http://www.theguardian.com/technology/2013/nov/10/teenagers-messenger-apps-facebook-exodus

[27] Rotta, R., Noack, A. (2011) Multilevel Local Search Algorithms for Modularity Clustering, Journal of Experimental Algorithms, Volume 16, Article no 2.3, DOI=http://doi.acm.org/10.1145/1963190.1970376.

[28] Samarati, P. (2001) Protecting Respondents' Identities in Microdata Release, IEEE Transactions on Knowledge and Data Engineering, vol. 13, no. 6, 1010-1027.

[29] Song, Y., Nobari, S., Lu, X., Karras, P., Bressan, S. (2011) On the Privacy and Utility of Anonymized Social Networks, Proceedings of the iiWAS'11, Ho Chi Minh City, Vietnam.

[30] Sweeney, L. (2002) K-anonymity: A Model for Protecting Privacy, International Journal on Uncertainty, Fuzziness, and Knowledge-based Systems, vol. 10, no. 5, 557 – 570.

[31] Truta, T.M., Campan, A., Gasmi, A., Cooper, N., Elstun, A. (2011) Centrality Preservation in Anonymized Social Networks, Proceedings of the International Conference on Data Mining (DMIN'11), Las Vegas, Nevada, USA.

[32] Truta, T.M., Campan, A., Ralescu, A.L. (2012) Preservation of Structural Properties in Anonymized Social Networks, Proceedings of the Collaborative Communities for Social Computing Workshop (CCSocialComp-2012), held in conjunction with CollaborateCom-2012, Pittsburgh, Pennsylvania, USA.

[33] Wasserman, S. Faust, K. (1994) Social Network Analysis: Methods and Applications, Cambridge: Cambridge University Press.

[34] Watts, D. J., Strogatz, S. H. (1998) Collective Dynamics of 'Small-World' Networks, Nature, Vol. 393, 440-442.

[35] Wu, W., Xiao, Y., Wang, W., He, Z., Wang, Z. (2010) K-Symmetry Model for Identity Anonymization in Social Networks, Proceedings of the Extending Database Technology Conference (EDBT), 111-122, DOI=http://doi.acm.org/10.1145/1739041.1739058.

[36] Yang, J., Leskovec, J. (2012) Defining and Evaluating Network Communities Based on Ground-Truth, Proceedings of the International Conference on Data Mining (ICDM).

[37] Zheleva, E., Getoor, L. (2007) Preserving the Privacy of Sensitive Relationships in Graph Data, Proceedings of the ACM SIGKDD Workshop on Privacy, Security, and Trust in KDD (PinKDD), 153-171.

[38] Zheleva, E., Terzi, E., Getoor, L. (2012) Privacy in Social Networks, Synthesis Lectures on Data Mining Series. Book published by Morgan and Claypool Publishers.

[39] Zhou, B., Pei, J. (2008) Preserving Privacy in Social Networks against Neighborhood Attacks, Proceedings of the IEEE International Conference on Data Engineering (ICDE), 506-515.