Preserving Structural Properties in Anonymization of Social Networks

Amirreza Masoumzadeh
School of Information Sciences
University of Pittsburgh
Pittsburgh, PA, USA
amirreza@sis.pitt.edu

James Joshi
School of Information Sciences
University of Pittsburgh
Pittsburgh, PA, USA
jjoshi@sis.pitt.edu

***Abstract*—A social network is a collection of social entities and the relations among them. Collection and sharing of such network data for analysis raise significant privacy concerns for the involved individuals, especially when human users are involved. To address such privacy concerns, several techniques, such as *k*-anonymity based approaches, have been proposed in the literature. However, such approaches introduce a large amount of distortion to the original social network graphs, thus raising serious questions about their utility for useful social network analysis. Consequently, these techniques may never be applied in practice. In this paper, we emphasize the use of network structural semantics in the social network analysis theory to address this problem. We propose an approach for enhancing anonymization techniques that preserves the structural semantics of the original social network by using the notion of roles and positions. We present experimental results that demonstrate that our approach can significantly help in preserving graph and social network theoretic properties of the original social networks, and hence improve utility of the anonymized data.**

I. Introduction

Social networks have increasingly attracted interest from different research communities such as academia, business, and even intelligence agencies. The focus of research on social networks is generally to identify structural properties and patterns in the data depending on the application of interest. With the advent of online social networks in recent years, while capturing and recording social interactions is becoming easier the concerns about the privacy of the individual users captured in such social networks also grow significantly. Such privacy concerns have direct influence on data handling practices and can become a significant burden on potentially fruitful collaboration and data sharing among organizations.

In order to reduce the risk of privacy violations by the exposure of privacy-sensitive information to unauthorized entities, it is important to anonymize the network data. Recent work on social network analysis show that naive anonymization of social network datasets such as simply removing personally identifiable information (PII) associated with the nodes of a network is not sufficient to fully preserve privacy [1][2][3]. Based on the topological structure of a network an adversary may be able to identify certain nodes by leveraging external background information that may be publicly available, e.g., over the Internet.

To cope with this problem, several researchers have proposed various anonymization techniques for social networks that can be broadly categorized into *perturbation* and *generalization* approaches. In a *perturbation* approach, the structure of the original network is slightly modified, usually by insertion/deletion of edges, to achieve a certain desired level of anonymity. The notion of *k*-anonymity has been primarily adopted from relational anonymization approaches for this purpose [4]. Alternatively, *generalization* approaches partition a social network into groups of nodes and replace them with hyper nodes; further, these methods only report the connectivity among hyper nodes and some associated properties such as the number of nodes and links within a hyper node.

Recent observations show that both these approaches severely suffer from a same problem: if data is anonymized up to an acceptable degree the results become highly distorted compared to the original networks, thus, severely affecting their utility for analysis purposes [3]. In order to use a social network anonymized by a generalization method it is needed to be reconstructed by randomly generating sub-structures in place of hyper nodes based on the reported hyper node properties in the results. Modifying links in the perturbation methods to fulfill the anonymization criteria (e.g., degree *k*-anonymity) also strongly affects the structure of the network. For instance, a node with a low centrality value may become of high centrality because of the introduction of many fake links to other nodes. Such a change can make any judgement made on the basis of the centrality of the nodes in the network invalid. The key problem related to these methods is that they usually focus on achieving the anonymization objectives and disregard the crucial need to preserve the original structural semantics of the network; hence, the outcome is a significant decrease in the utility of the results.

In this paper, we consider such structural semantics in the anonymization process by using concepts from the social network analysis theory [5]. In particular, we leverage the notion of structural roles and positions as the key entities to enhance existing perturbation techniques so that the original structural semantics are preserved. As we demonstrate in this paper, this approach shows significant improvements in maintaining the structural measurements of the social networks such as network diameter, betweenness centrality, clustering

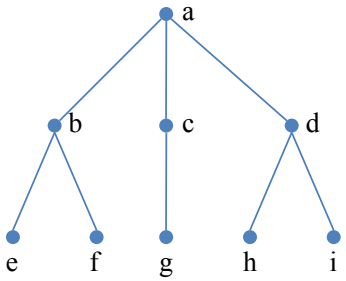


Fig. 1: A Social Network of Managers

coefficient, etc., all of which have direct effect on the usefulness of the anonymization results. The key contributions of the proposed work are as follows.

- We provide a formal approach towards preserving role structure in social networks during perturbation. To the best of our knowledge, this is the first attempt in the literature that leverages such network theoretic properties to lessen the negative effects of perturbation on the structure of a social network.
- Based on the proposed formalism, we outline an enhancement approach that can be easily applied to most of the key perturbation techniques.
- We present experiments on enhancing two specific algorithms proposed in the literature and demonstrate the improvements on the utility of the anonymized networks. The results show very encouraging results on preserving structural properties of the social network, compared to the original version of the algorithms.

The rest of the paper is organized as follows. In Section II, we present preliminary concepts and techniques for our approach. In particular, we define the notion of roles and positions in a social network and present the conceptual equivalency approaches to classify actors. Then, we review an algorithm to identify such equivalency classes, and present required modification to adapt it for undirected social networks. We also outline a generalized perturbation algorithm that is later used as reference to apply the enhancements. In Section III, we propose a formal approach to preserve role structure during social network perturbation, and show how such an approach can be used to enhance a typical perturbation technique based on the algorithm outlined in Section II. We empirically evaluate enhancement of structural network properties in two major perturbation techniques by using our approach in Section IV. In Section V, we review the related literature, and subsequently conclude the paper in Section VI.

II. Preliminaries

A. Roles and Equivalence in Social Networks

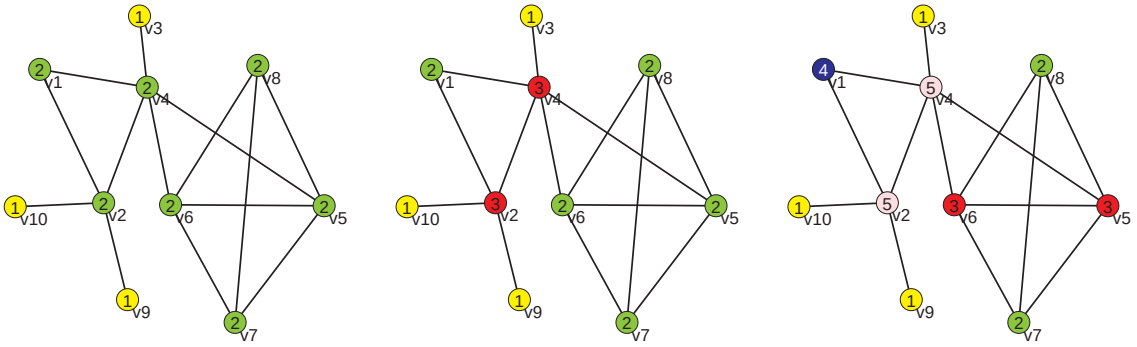
Roles and positions are very helpful in representing the structure of a social network. Figure 1 shows an example social network of managers in a small company in which vertices represent managers and edges show direct contact among them (adopted from [5]): manager *a* has direct contact with managers *b*, *c*, and *d*, manager *b* has direct contact with managers *e* and *f*, etc. We can intuitively identify three roles in this social network: *top manager* (*a*), *middle manager* (*b*, *c*, and *d*), and *line manager* (*e*, *f*, *g*, *h*, and *i*). Roles can indicate many structural properties of social networks such as centrality measures. In our example, the actors with the *middle manager* role have a lower centrality than the actor with the *top manager* role, and higher centrality than the actors with the *line manager* role.

There are three major approaches to classify actors in a network into their social positions based on the relations among them. Each approach defines graph theoretic properties that sets of actors must have in order to be considered equivalent in terms of roles they play. The equivalence classes formed this way represent positions [5][6]. *Structural equivalence* is the simplest approach, which requires each two actors in the same class to have identical ties with identical other actors. For instance, in Figure 1, *e* and *f* are structurally equivalent, so are *h* and *i*; no other pair of structurally equivalent actors exists. The set of equivalency classes is $\{\{a\},\{b\},\{c\},\{d\},\{e,f\},\{g\},\{h,i\}\}$. *Automorphic equivalence* relaxes the structural equivalence requirement by requiring actors in the same position to have identical ties with different sets of actors that play the same role in relation to that position. The set of automorphic equivalency classes in Figure 1 is $\{\{a\},\{b,d\},\{c\},\{e,f,h,i\},\{g\}\}$. *Regular equivalence* is the least restrictive approach. Actors are regularly equivalent if they have same kind of relations with actors that are also regularly equivalent. This results in $\{\{a\},\{b,c,d\},\{e,f,g,h,i\}\}$ as the set of equivalency classes of Figure 1.

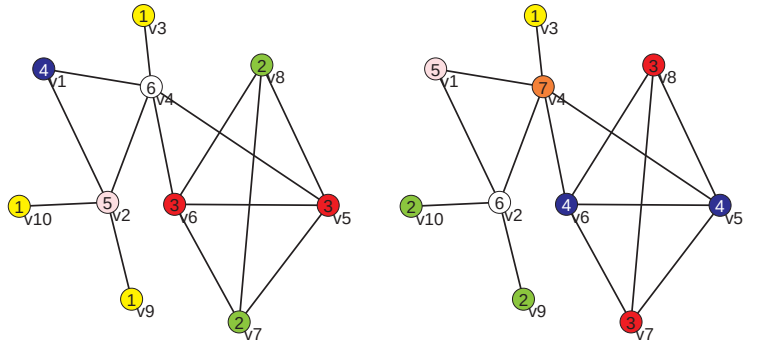
The above three approaches were represented in decreasing order of restrictiveness. The less restrictive the approach is, the more populated the equivalency classes become. In this paper, we use the regular equivalence, which is the least restrictive concept. This makes the perturbation enhancement process that we propose in this paper more flexible and effective.

B. Identifying Roles Using CATREGE

CATREGE [7] is a popular algorithm for computing regular equivalence of categorical data that also provides an intuitive role similarity measure. Although the key assumption in categorical network data is the existence of different edge types, CATREGE can also work perfectly for non-categorical data (that is the concern of this paper). CATREGE requires as input a multiplex adjacency matrix of the target network. The values of such a matrix are categorical codes that index each unique combination of input relations (and their inverses) that connect each pair of nodes. For instance, for a single directed relation R, the possible values are 1 (if iRj but not jRi), 2 (if jRi but not iRj), 3 (if iRj and jRi), and 0 (if not iRj and not jRi). Given a multiplex matrix, CATREGE iteratively verifies that pairs of nodes that were equivalent in the previous iteration have the same type of multiplex relations with their neighbors. If not, they are marked as non-equivalent. All actors are assumed equivalent prior to the first iteration. The procedure is repeated until there is no change in



(a) Initial State (b) Iteration 1 (c) Iteration 2



(d) Iteration 3 (e) Iteration 4

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
v_1	5	1	0	1	2	2	2	2	0	0
v_2	1	5	0	3	1	1	1	1	0	0
v_3	0	0	5	0	0	0	0	0	4	4
v_4	1	3	0	5	1	1	1	1	0	0
v_5	2	1	0	1	5	5	2	2	0	0
v_6	2	1	0	1	5	5	2	2	0	0
v_7	2	1	0	1	2	2	5	5	0	0
v_8	2	1	0	1	2	2	5	5	0	0
v_9	0	0	4	0	0	0	0	0	5	5
v_{10}	0	0	4	0	0	0	0	0	5	5

(f) (Non-normalized) Similarity Matrix

Partition Color Codes: 1=Yellow, 2=Green, 3=Red, 4=Purple, 5=Pink, 6=White, 7=Orange.

Fig. 2: A Sample Execution of the Modified CATREGE Algorithm

equivalencies compared to the previous iteration. The extent of regular equivalence between two actors can be obtained by counting the number of iterations it takes them to split into different partitions. This value can be normalized by dividing it by the total number of iterations. The result is a similarity measure in the range $[0, 1]$.

In this paper, we deal with non-categorical (single-type edge), undirected social networks in this paper. Employing CATREGE for such social networks results into an uninteresting regular equivalence: all the actors will be classified in the same equivalency class. This is because only one bundle of relations exists here, and therefore the multiplex matrix would only constitute of zeros and ones. In order to tackle this issue, instead of starting with the full partition that has all the actors in the same equivalence class, we initialize the algorithm with two partitions: nodes with only one neighbor are grouped into one partition, separated from the rest of the nodes in the other partition. In other words, we consider the perimeter nodes in a network more regularly equivalent to each other, and less to the others that fall inside the network. Note that each iteration of the CATREGE algorithm will further classify the actors in each of the two initial partitions.

Figure 2 illustrate the execution of our modified version of CATREGE on a small network. In each iteration, vertices within the same partition are marked with the same color (number). Note that partition colors (numbers) are just to indicate equivalent actors in one iteration and do not carry any other semantics. In the initial state (Figure 2a), vertices v_3, v_9, and v_{10} are colored yellow, and all the others are colored green. Figure 2b illustrates the resultant partitions after the first iteration of CATREGE. Since in the previous step, the yellow vertices were all connected to the green vertices, they will not separate in this iteration. However, the previously green vertices become divided into two partitions: the ones that were only connected to greens, and the ones that were connected to both yellows and greens. If we continue the procedure, the final result is obtained after iteration 4 (Figure 2e); further iterations will not change the partitions. Figure 2f shows the (non-normalized) extent of regular equivalence between pairs of actors. For instance, the similarity value for v_1 and v_2 is 1, because they were separated after the first iteration. Analogously, the similarity value for v_5 and v_7 is 2, because they were separated after the second iteration. If two vertices are eventually remain equivalent their similarity will be maximum number of steps (e.g., 5 for v_5 and v_6). A normalized version of this similarity matrix can be obtained by dividing the elements by 5.

C. Overview of Perturbation Techniques

In this section, we provide an abstract overview of the perturbation algorithms for social network anonymization.

Later in Section III, we present the details of the proposed enhancements to these generalized algorithms towards preserving structural semantics in the anonymized data. Perturbation techniques output a graph with modified edge structure compared to the original one, which satisfies a specific *anonymization criteria*. These techniques typically follow a greedy iterative approach, which can be abstractly expressed as in Algorithm 1. In each iteration, Algorithm 1 selects an edge to be inserted/deleted using a heuristic which depends on the specific technique. The iterations continue until the graph is considered anonymized according to the anonymization criteria. The algorithm aborts if the anonymization criteria cannot be achieved. Different anonymization techniques have different anonymization criteria. In the *random perturbation* technique [2], the goal is to simply delete m edges randomly and then insert m random edges. In *k-anonymity-based* approaches (e.g., [8], [9], [10], [11]), the goal is, for instance, to achieve a graph with k-anonymous vertex degrees (such as *Supergraph*[8] or *Union-Split*[10]).

Algorithm 1 Iterative Edge Perturbation Algorithm

```
Start from the original graph
repeat
    if an edge should be inserted then
        Choose non-existent edge {u, v} to be inserted
        Insert {u, v}
    end if
    if an edge should be deleted then
        Choose existing edge {u, v} to be deleted
        Delete {u, v}
    end if
until anonymization criteria is achieved
```

The *Greedy Swap* algorithm proposed in [8] includes an optimization phase to select a group of edge changes in the graph in each iteration, which results in a slightly different algorithm scheme (see Algorithm 2). The algorithm first creates a random anonymized graph based on the k-anonymous degree sequence of the graph. In each iteration, every pair of edges in a subset of existing edges is examined to be selected for a *swap*. In a *swap* operation, a pair of edges are replaced with another pair using the same end nodes. Two swap options are considered for a pair of edges $\{\{u, v\}, \{u', v'\}\}$: either $\{\{u, u'\}, \{v, v'\}\}$, or $\{\{u, v'\}, \{u', v\}\}$. Such swaps do not change vertex degrees thus ensuring the already-established degree k-anonymity. A *gain value* is calculated for each swap option, and the swap with maximum (positive) gain is selected. In [8], the authors calculate the *gain value* as the increment of edge overlap (intersection) between the interim and the original graph. Performing the swap with maximum gain at each iteration would greedily make the anonymized graph more structurally similar to the original one.

Algorithm 2 The Greedy Swap Algorithm

```
Create an anonymized random social network
repeat
    Select log(|E|) of existing edges randomly
    for all Pairs of selected edges {u, v} and {u', v'} do
        Calculate the gain value considering swapping the
        pair either with {u, u'} and {v, v'}, or {u, v'} and
        {u', v}
    end for
    Perform the swap with maximum gain (if any)
until No edge swap is performed
```

III. Preserving Structure in Perturbation Techniques

In this section, we formally define the notion of roles and related concepts in the context of undirected social networks; we adopt some definitions from [12]. Then, we present a formal approach to preserve the role structure during graph perturbation. Finally, we extend the algorithms outlined in Section II-C, using the proposed structure-preserving approach.

A. Preliminaries

We define a social network as an undirected graph $G\langle V, E\rangle$, where the set of vertices V represents the actors in the network, and the set of edges $E \subseteq \{\{u, v\} | u, v \in V\}$ represent the links between actors in V.

Definition 1 (Role Assignment): A role assignment for network $G\langle V, E\rangle$ is a surjective function $\Phi : V \to R$, defined for every member of V, where R is a set of roles.

A role assignment partitions actors into equivalency classes. Two actors are considered *equivalent* if they are assigned the same role: $\forall u, v \in V; u \equiv_\Phi v \Leftrightarrow \Phi(u) = \Phi(v)$. In other words, a role assignment is a projection of an equivalence relation. Of our particular interest is the *regular equivalence*. The following definition captures the relations between actors.

Definition 2 (Neighbor Role Set): $\Gamma_\Phi : V \to 2^R$ is a function that maps an actor in network $G\langle V, E\rangle$ to the roles of its neighbors according to role assignment Φ, i.e., $\Gamma^G_\Phi(u) = \{\Phi(v) | \{u, v\} \in E\}$.

Recall that regularly equivalent actors (actors that are assigned the same role) must have the same kind of relations with other regularly equivalent actors. A role assignment that projects a regular equivalence relation is defined as follows.

Definition 3 (Regular Equivalence Role Assignment): A role assignment $\Phi : V \to R$ projects a regular equivalence for actors in $G\langle V, E\rangle$ if and only if

$$\forall u, v \in V, \Phi(u) = \Phi(v) \Rightarrow \Gamma_\Phi(u) = \Gamma_\Phi(v).$$

We refer to this as *RE-role assignment* in the rest of the paper. We observe that despite regular equivalence being the least restrictive approach in identifying positions, synthetic algorithms for computing it such as CATREGE (which use no external semantics other than the network structure) result in very low-populated equivalency classes. However, as we discuss later, our perturbation enhancement approach relies highly on the existence of alternatives same-role actors. We tackle this issue by using the extent of the (dis)similarity

between actors. We abstractly define a dissimilarity measure for roles as follows.

Definition 4 (Regular Equivalence Role Dissimilarity): $\Delta_\Phi : V \times V \rightarrow [0,1]$ is a role dissimilarity function for actors of network $G\langle V, E\rangle$ corresponding to role assignment Φ where $\Delta_\Phi(u,v) = 0$ implies actors u and v have the same role ($\Phi(u) = \Phi(v)$), and $\Delta_\Phi(u,v) = 1$ implies actors u and v have completely dissimilar roles.

The actual values of the function can depend on the role identification scheme used. In this work we use the similarity measure provided by the CATREGE algorithm (Section II-B), and subtract it from 1 to obtain the dissimilarity values between roles. Subsequently, we are interested in a dissimilarity measure between two sets of roles, based on the dissimilarity measure we have for individual pairs of roles; we define it as follows.

Definition 5 (Regular Equivalence Role Set Dissimilarity): Let $S \subseteq R$ and $S' \subseteq R$ be two sets of roles. The regular equivalence dissimilarity between S and S', written as $\Lambda(S, S')$, is calculated as follows:

$$\Lambda(S,S') = \frac{\frac{\sum_{x \in S} \sqrt[|S'|]{\prod_{y \in S'} \Delta(x,y)}}{|S|} + \frac{\sum_{y \in S'} \sqrt[|S|]{\prod_{x \in S} \Delta(x,y)}}{|S'|}}{2}$$

The above formula essentially calculates the (asymmetric) dissimilarities of S to S', and S' to S, and then takes the average to compute an overall (symmetric) dissimilarity between S and S'. The dissimilarity of S to S' (the first expression in the numerator) is calculated as follows. For every role x in S, the product of its dissimilarities with all roles in S' is calculated, and its $|S'|^{\text{th}}$ root is taken. This gives us an overall dissimilarity value between x and roles in S'. If one of the roles in S' is the same as x the result would be zero; otherwise the dissimilarity values for each will be effective in the result. The average of all such dissimilarities for all the roles in S is considered as the dissimilarity of S to S'. The dissimilarity of S' to S is calculated in a similar fashion.

B. Formalizing Role Structure Preservation

Our intuition is that preserving the role structure in a network in the anonymization process would essentially preserve the network structural properties that a social network analyzer may be looking for in the anonymized network. To be more specific, our goal is to ensure that an RE-role assignment in the original network is applicable to its edge-perturbed version as well. However, modifications to the edge structure of a network during perturbation can easily thwart this goal. The following theorem captures a sufficient condition for preserving an RE-role assignment in the edge perturbation process.

Theorem 1: Let $G'\langle V, E'\rangle$ be an edge-perturbed version of network $G\langle V, E\rangle$. An RE-role assignment Φ for G is also an RE-role assignment for G' if

$$\forall u \in V \ [\Gamma_\Phi^{G'}(u) = \Gamma_\Phi^{G}(u)] \tag{1}$$

Proof: The proof is straightforwardly implied from Definition 3 and condition (1). For every u and v where $\Phi(u) = \Phi(v)$, by Definition 3 we have $\Gamma_\Phi^G(u) = \Gamma_\Phi^G(v)$. Considering condition (1) we have $\Gamma_\Phi^{G'}(u) = \Gamma_\Phi^G(u) = \Gamma_\Phi^G(v) = \Gamma_\Phi^{G'}(v)$, and hence $\Gamma_\Phi^{G'}(u) = \Gamma_\Phi^{G'}(v)$. This is essentially the sufficient condition for Φ to be an RE-role assignment for G'. ■

The above theorem simply states that keeping the neighbor role sets of actors in a network intact in the anonymization process will preserve an RE-role assignment. As an edge perturbation algorithm involves a series of edge insertions/ deletions, the above condition can be further captured with regards to the set of inserted or deleted edges as in the following theorem.

Theorem 2: Let $G'\langle V, E'\rangle$ be an edge-perturbed version of network $G\langle V, E\rangle$. An RE-role assignment Φ for G is also an RE-role assignment for G' if the following conditions are met

$$\forall \{u,v\} \in E_i \ \exists \{u,v'\} \in E \ [\Phi(v) = \Phi(v')] \tag{2}$$

$$\forall \{u,v\} \in E_d \ \exists \{u,v'\} \in E' \ [\Phi(v) = \Phi(v')] \tag{3}$$

where sets $E_i = E' \backslash E$ and $E_d = E \backslash E'$ represent inserted and deleted edges, respectively.

Proof: Since the same role assignment Φ is considered for both G and G', any difference between $\Gamma_\Phi^G(u)$ and $\Gamma_\Phi^{G'}(u)$, for any actor u, can only be the result of either insertion or deletion of an edge adjacent to u. For an inserted edge $\{u,v\}$, by condition (2) we have $\exists \{u,v'\} \in E[\Phi(v) = \Phi(v')]$ and therefore $\Phi(v) = \Phi(v') \in \Gamma_\Phi^G(u)$, i.e., an inserted edge would not affect the neighbor role set of an actor. For a deleted edge $\langle u,v\rangle$, by condition (3) we have $\exists \{u,v'\} \in E'[\Phi(v) = \Phi(v')]$ and therefore $\Phi(v) = \Phi(v') \in \Gamma_\Phi^{G'}(u)$, i.e., a deleted edge would not affect the neighbor role set of an actor. These suggest

$$\forall u \in V \ [\Gamma_\Phi^{G'}(u) = \Gamma_\Phi^{G}(u)]$$

which is sufficient condition for Φ to be an RE-role assignment for G' according to Theorem 1. ■

C. Preserving Structure in Iterative Edge Perturbation Algorithms

We use Theorem 2 to extend and enhance the iterative edge perturbation techniques represented by Algorithm 1 as follows. After selecting an edge for insertion, the insertion is performed only if it conforms to condition (2). For this purpose, line 5 of the algorithm should be replaced with the following.

if $\exists \{u,v'\} \in E \ [\Phi(v) = \Phi(v')] \wedge \ \exists \{u',v\} \in E \ [\Phi(u) = \Phi(u')]$ **then**
  Insert $\{u,v\}$
end if

This checks if there exists vertex v' in u's neighborhood with the same role as v's, and that there exists vertex u' in v's neighborhood with the same role as u's. If the checks fail the insertion decision is ignored. Analogously, a deletion should be allowed if it conforms to condition (3). As per Theorem (2), such a modified version of Algorithm 1 will preserve an RE-role assignment for the graph in each iteration. Therefore, an RE-role assignment for the original social network graph will be valid for its final edge-perturbed version.

Although theoretically sound, the above-mentioned conditions may not perform well in practice. The key issue, as briefly mentioned in Section III-A, is that algorithms such as CATREGE identify very small number of actors with the same role. Therefore, when inserting/deleting edge $\{u, v\}$ there is a low probability of finding an actor with same role as v's in u's neighborhood and vice versa, which is required by the above conditions. In order to overcome this limitation, we use a relaxed version of the conditions in Theorem (2), by using a threshold on RE-role dissimilarity between roles instead of checking the exact role match. Algorithm 3 provides pseudocode for the enhanced version of the iterative edge perturbation approach. Here, $\delta \in [0, 1]$ is a constant that specifies the allowed extent of non-perfect role matching.

Algorithm 3 RE-Enhanced Iterative Edge Perturbation Algorithm

```
Start from the original graph
repeat
  if an edge should be inserted then
    Choose non-existent edge {u, v} to be inserted
    Let G⟨V, E⟩ be the current graph
    if ∃{u, v'} ∈ E [Δ_Φ(v, v') < δ] ∧ ∃{u', v} ∈ E [Δ_Φ(u, u') < δ] then
      Insert {u, v}
    end if
  end if
  if an edge should be deleted then
    Choose existing edge {u, v} to be deleted
    Let G'⟨V, E'⟩ be the graph after deleting edge {u, v}
    if ∃{u, v'} ∈ E' [Δ_Φ(v, v') < δ] ∧ ∃{u', v} ∈ E' [Δ_Φ(u, u') < δ] then
      Delete {u, v}
    end if
  end if
until anonymization criteria is achieved
```

D. Preserving Structure in Greedy Swap Algorithm

As mentioned in Section II-C, the *greedy swap* algorithm follows a different overall procedure than most of the other perturbation approaches. Hence, we need a different approach to enhance it for preserving role structure. We propose to substitute the *gain function* in Algorithm 2 with a new similarity gain measure. Note that in Algorithm 2 the gain measure captures increase in edge overlap. The new gain function is intended to measure how much each of the involved vertices in an edge swap is closer (more similar) to their corresponding original states in terms of role structure. Recall from Theorem 1 that the neighbor role set of an actor acts as an important factor in preserving its role. Hence, we consider it as the main clue for calculating such a similarity gain measurement.

There are four vertices involved in a swap of a pair of edges $\{u, v\} and \{u', v'\}$. For vertex u, in the i^{th} iteration in Algorithm 2, let $\Gamma_{\Phi}^{G}(u)$ be its neighbor role set in the original network, $\Gamma_{\Phi}^{G_i}(u)$ be its neighbor role set in the interim network, and $\Gamma_{\Phi}^{G_{i+1}}(u)$ be its neighbor role set in the next state of the interim network if the swap is performed. The objective of optimizing based on role similarity gain is to make $\Gamma_{\Phi}^{G_{i+1}}(u)$ more similar than $\Gamma_{\Phi}^{G_i}(u)$ to $\Gamma_{\Phi}^{G}(u)$. We calculate such a gain by measuring the *decrease in dissimilarity* between the neighbor role set in the original and interim networks. Thus the total role similarity gain of a swap is calculated as follows.

$$\frac{\sum_{x \in \{u,v,u',v'\}} [\Lambda(\Gamma_{\Phi}^{G}(x), \Gamma_{\Phi}^{G_i}(x)) - \Lambda(\Gamma_{\Phi}^{G}(x), \Gamma_{\Phi}^{G_{i+1}}(x))]}{4}$$

IV. Experimental Result

A. Setup

We use the following undirected network dataset:

- *jazz*: a social network of jazz musicians[1] ($|V| = 198$ and $|E| = 5484$)

We have implemented the original and the RE-enhanced versions of the *random perturbation* and the *greedy swap* techniques in Java. We used our version of CATREGE according to the descriptions in Section II-B to generate regular equivalency classes and the corresponding dissimilarity matrix. These served as extra inputs to the enhanced algorithms. The algorithms were run for 10 rounds and the average of the measurements are reported. We measure the following graph and social network theoretic parameters:

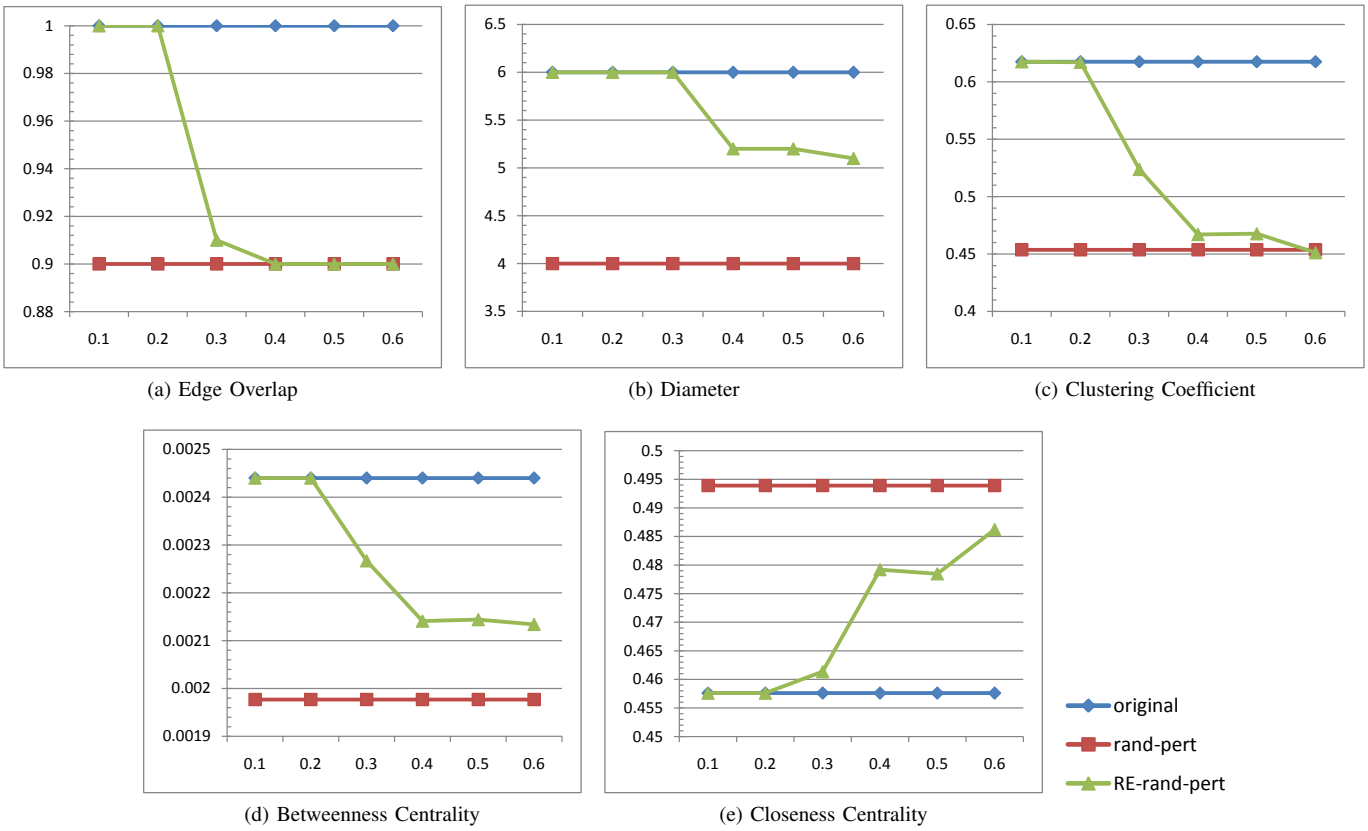
- *Edge overlap*: the proportion of edges in the anonymized network that overlap with the original network.
- *Diameter*: the longest shortest path between any pair of vertices in the graph.
- *(Average) Clustering Coefficient*: the proportion of links between the vertices within one vertex's neighborhood to the number of links that could possibly exist between them.
- *(Average) Betweenness Centrality*: the proportion of all shortest paths between that pass through a vertex.
- *(Average) Closeness Centrality*: the mean shortest path between a vertex and all other vertices reachable from it.

The closer a perturbed network's measurements are to the original network's, the more preserved its structure becomes, and hence better is the utility of such an anonymized network.

B. Evaluating Enhancement of The Random Perturbation Technique

We evaluate the proposed enhancement method for edge perturbation presented in Section III-C by running the random perturbation algorithm [2][13] against its RE-enhanced version. As suggested in [2], in order to achieve sufficient anonymization, we have choosen to delete randomly 10% of the original edges and insert back randomly the same number of edges. Figure 3 illustrates the results for the jazz network, tested over different values of δ (the dissimilarity threshold of considering items to be regularly equivalent). As seen in Figure 3a, for $\delta = 0.1$ and $\delta = 0.2$, the resulting RE version fully overlap with the original graph.

[1] Available at http://deim.urv.cat/~aarenas/data/welcome.htm



(a) Edge Overlap (b) Diameter (c) Clustering Coefficient

(d) Betweenness Centrality (e) Closeness Centrality

Fig. 3: Effects of Random Perturbation and Its RE-Enhanced Variation on Structural Properties of Jazz Network ($0.1 \leq \delta \leq 0.6$)

This means that the strict thresholding on the enhancement in fact has allowed only those original edges which were randomly removed to be added back, which effectively defeats the anonymization purpose. However, results for $\delta \geq 0.3$ seem to indicate that sufficient anonymization has been performed, i.e., the edge overlap is approximately the same as in the non-enhanced version. The results show significant improvement in the network measurements, where our RE-enhanced approach almost outperforms the original algorithm for all δ values (Figure 3b-3e). Increasing the value of δ intuitively reduces the effectiveness of the enhancement procedure, as more non-perfect perturbations (with regard to role structure) are allowed. Therefore, identifying a suitable threshold that balances the utility enhancement while keeping the anonymization property (here, the randomness of inserted edges, which is measured by edge overlap) is necessary. A value of $\delta = 0.3$ seems to provide such a desirable tradeoff for the jazz dataset.

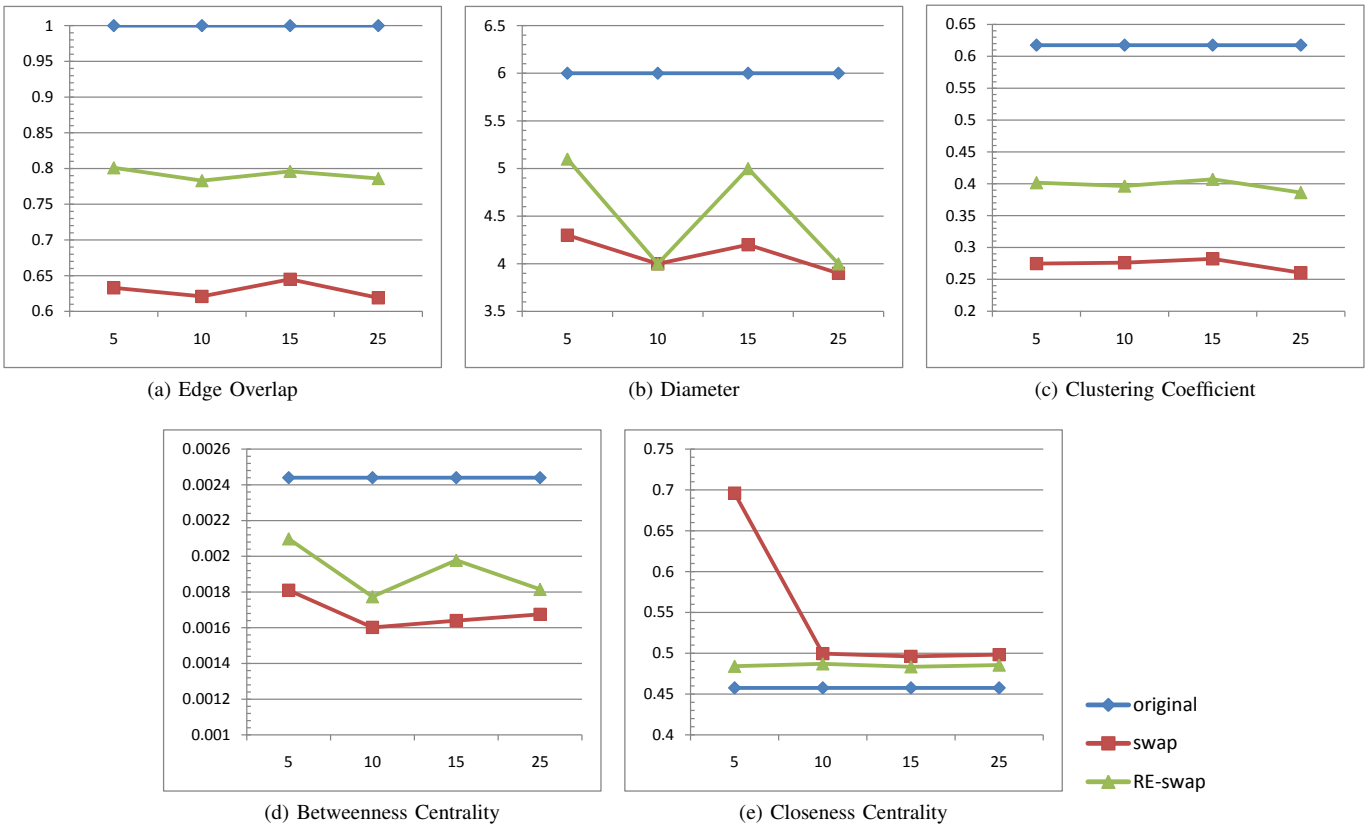
C. Evaluating Enhancement of The Greedy Swap Technique

We evaluate the performance of our proposed enhancement for the greedy swap method in Section III-D, as the measurement results of the jazz network for different k values (the k parameter in k-anonymity) are illustrated in Figure 4. The edge overlap (Figure 4a) is constantly improved to about 0.8 in the RE-enhanced algorithm compared to 0.6 in the original algorithm. Note that unlike the random perturbation, here the anonymization quality is not reflected by the edge overlap. The other four measured properties (Figure 4b-4e) are also often considerably closer to the original graph in the RE-enhanced results than the original greedy swap algorithm results, regardless of the value of k. Note that the increasing values of k in these figures is not supposed to correlate with increase or decrease in property measurements, as different values of k may impose different structural changes, depending on the original network structure.

V. Related Work

A. Privacy Risks

Naive anonymization of social networks, i.e., replacing true node identifiers with random ones has been shown to be ineffective, similar to observations on anonymization techniques in relational database literature [4]. While in the case of relational data a subset of an entity's attributes may help with unique identification (quasi-identifier), in the case of social networks the connectivity of an entity and its surrounding entities in the network can be revealing without a need for explicit data attributes (as in relational data). Backstrom et al. discovered a family of active/passive attacks that work based on uniqueness of some small random subgraphs embedded in a network [1]. In active attacks, an adversary chooses a target set of users, creates small number of new users, and then creates a pattern of links among the newly created



(a) Edge Overlap (b) Diameter (c) Clustering Coefficient

(d) Betweenness Centrality (e) Closeness Centrality

Fig. 4: Effects of Greedy Swap and Its RE-Enhanced Variation on Structural Properties of Jazz Network ($5 \leq k \leq 25$)

users, in the way that it stands out in the naively anonymized version of the network. In passive attacks, existing users of a network collude to re-identify certain nodes connected to them, based on the fact that such a small coalition of friends can uniquely identify the subgraph of their coalition by exchanging structural information.

Hay et al. study the extent of node re-identification based on structural information [2][14]. They define *k-candidate anonymity* for a structural query based on the notion of *k*-anonymity [4]. Three types of structural queries are considered as adversary background knowledge. Vertex refinement queries are iteratively defined and capture degrees of different distant levels of neighbors around a target. Subgraph queries capture a less complete structure surrounding a target than vertex refinement based on known edges to the adversary. Hub fingerprints express a target's distance from certain hubs in the network. Their experiments on real, naively anonymized social networks show significantly low *k*-candidate anonymity for such background knowledge queries.

Narayanan et al. propose a different attack approach that relies on input of an auxiliary, overlapping, probably publicly available social network without any assumption about structural background knowledge of an adversary [3]. Empirical evaluation of their approach shows that a third of users who have accounts both on Twitter and Flickr can be re-identified in the anonymous Twitter graph with a low error rate.

B. *Anonymization by Generalization*

The non-naive social network anonymization approaches in the literature can be categorized into two groups: graph generalization and graph perturbation. In graph generalization techniques, the network graph 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.

Hay et al. propose a generalization approach in which actors are partitioned into groups of size at least k (a k-anonymity approach), and edge densities within and between partitions are reported [14]. Their algorithm optimizes fitness to the original network via a maximum likelihood approach. Zheleva et al. study re-identification of sensitive links in a network that may disclose node attributes and nonsensitive links [15]. Accurate probabilistic model of predicting sensitive edges based on observing nonsensitive edges is assumed as adversary background knowledge. The authors consider different anonymization approaches including combination of node attribute anonymization and partial edge removal, and graph generalization which avoid disclosure of exact nonsensitive edge structure. Campan et al. also follow a graph generalization approach similar to, but more detailed than the approach in [15]. They provide formal information loss measurement due to attribute generalization and structural generalization, and use them to greedily optimize their proposed clustering

anonymization.

In order to use a generalized social network for analysis purpose, one should sample a random graph in accordance with the reported generalized properties. Although such a network may maintain some local structural properties of the original network, much of high-level graph structure is lost [10], which impacts negatively the utility of results.

C. Anonymization by Perturbation

In graph perturbation techniques, the network graph is (slightly) modified to meet desired privacy requirements. This is usually carried out by inserting and/or deleting graph edges. Although, theoretically, perturbation can be introduced to graph nodes (i.e., network actors) as well, it is not considered plausible because of adverse effects on the dataset.

Hay et al. propose a random perturbation approach, in which a sequence of m edge deletions followed by m edge insertions[2]. Assuming an adversary needs to consider the set of possible worlds implied by m deletions/insertions, the authors reason that it could be intractable for an attacker to achieve exact identification. However, this cannot guarantee that the adversary will not succeed in (sufficiently accurate) identification of selected individuals. Ying et al. analyze the privacy protection provided by the random perturbation approach [13]. They formulate the confidence of an adversary in identifying a node in the anonymized network based on the degree of the target as background knowledge.

Liu et al. propose an edge perturbation approach that provides *k*-anonymity for vertices based on their degrees. Initially a *k*-anonymized degree sequence for the graph is constructed, in which there exist at least k nodes of each degree and the total degree difference between the anonymized and the original degree sequence is minimum. Then the problem reduces to realizing a graph with the anonymized degree sequence from the original graph. They propose two different algorithms to solve it. The *Supergraph* algorithm greedily perturbs the original graph until it reaches to the target anonymized degree sequence. Since such a greedy algorithm cannot guarantee an answer, a probing scheme is proposed by the authors that retries the procedure with slight modification of the degree sequence, until an anonymized graph is realized. The *Edge Swap* algorithm starts by constructing a random graph based on the anonymized degree sequence. It then modifies the graph to maximize its overlap with the original graph, while preserving the anonymized degree sequence.

Thompson et al. propose a *k*-anonymity-based two phase clustering and perturbing approach [10]. Vertices are clustered first into groups of size of at least k, and then edges are greedily inserted/deleted so that each vertex is anonymous to the vertices in its corresponding cluster. As the adversary background knowledge criteria, they consider an approach similar to vertex refinement queries for zero and one-level neighborhoods [2]. As the parameter for clustering, the vertex degree, and a linear combination of a vertex and its neighbors' degree are used for zero and one-level neighborhood, respectively. They propose two alternative clustering algorithms for this purpose. *Bounded t-Means* is a constrained version of traditional *k*-means algorithm that limits the number of nodes in a cluster to k. *Union-Split* is an alternative agglomerative clustering algorithm. It starts from each node as a cluster and in each step joins two nearest clusters. If the joint cluster size is more than $2k$ it is split into two cohesive clusters, each of size at least k. The iteration continues until all clusters have k or more members. Although the clustering approach seems promising, unfortunately their proposed greedy perturbation algorithm based on clusters does not guarantee an answer, despite the authors' claim.

Zhou et al. propose a scheme to *k*-anonymize one-level neighborhood of every vertex in a graph that vertices carry labels as attributes. The neighborhood of each vertex is made isomorphic with at least $k-1$ other neighborhoods. The isomorphic perturbation process greedily minimizes information loss in generalizing labels and inserting edges during perturbation. In this approach, since every node's neighborhood considered independently in the *k*-anonymization process, perturbation of one neighborhood can easily invalidate the *k*-anonymity of an already anonymized node that falls inside the neighborhood. This results into recurring anonymization of such nodes, and therefore, an inefficient process with high graph distortion. He et al. propose a different neighborhood anonymization scheme [11]. They first partition the graph into local structures, and ignore the inter-partition edges. Then the neighborhoods are formed in the groups of size at least k and the neighborhoods in each group are made isomorphic to each other using edge perturbation. In the last step the previously ignored inter-partition edges are put back in the way that it does not violate the isomorphisms. The authors leverage existing graph algorithms for local structure partitioning and grouping purpose. Although acceptable performance regarding preserving structural properties have been reported in the work, it is not clear if the results are generalizable as no comparison is provided with the related approaches in the literature. Furthermore, we believe that making every k-grouped partitions isomorphic and inserting back inter-partition edges in an isomorphic-preserving manner as adopted in [11] will create a very symmetric structure; considering the need for insertion of about k^2 edges per original inter-partition edge, the result does not seem to maintain well its original structural properties in general.

We take an enhancement approach in this paper, rather than offering alternatives to the perturbation algorithms in the literature. We believe that our approach is applicable to most of the perturbation techniques, and empirically show improvements over the original versions of some perturbation algorithms.

VI. Conclusions and Future Work

Privacy is a huge concern when sharing social network datasets. Existing social network anonymization techniques usually do not perform well in terms of maintaining the utility of the final outcome; the distortions on the original datasets have drastic effects on their analysis. In this paper, we propose

the first approach to preserve structural properties of social networks in an anonymization process, using the concept of roles in a network. We have presented generalized enhanced algorithms for social network perturbation. Our experimental results show significant improvement in preserving structural semantics of networks using our approach compared to the original techniques in the literature.

There is a plausible vulnerability with our enhancement approach if an attacker can leverage the role structure of the network as background knowledge. Ideally the attacker can link the actors with the same role in the original and the anonymized network, and therefore potentially defeat the anonymization scheme. We believe that this is a reasonable concern when using perfect role structure preservation, for instance, as suggested by Theorem 2. However, as mentioned later in Section III, we need to use dissimilarity measures for role matching instead of perfect role match. We believe that this will introduce enough noise to the structure that will help preventing such an attack strategy. Note that even few imperfect changes in the network structure would result in completely different role structure identified by Algorithms such as CATREGE. We will investigate more formally such attacks as a future work. Moreover, we plan to conduct experiments on a more diverse set of datasets and techniques.

ACKNOWLEDGMENT

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

REFERENCES

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

[2] M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava, "Anonymizing social networks," University of Massachusetts Amherst, Tech. Rep. 07-19, 2007.

[3] A. Narayanan and V. Shmatikov, "De-anonymizing social networks," in *2009 30th IEEE Symposium on Security and Privacy*, vol. 0. Los Alamitos, CA, USA: IEEE, August 2009, pp. 173–187.

[4] L. Sweeney, "k-anonymity: a model for protecting privacy," *Int'l Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[5] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

[6] D. Knoke and S. Yang, *Social Network Analysis (Quantitative Applications in the Social Sciences)*, 2nd ed. Sage Publications, Inc, November 2008.

[7] S. Borgatti, "Two algorithms for computing regular equivalence," *Social Networks*, vol. 15, no. 4, pp. 361–376, December 1993.

[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*. New York, NY, USA: ACM, 2008, pp. 93–106.

[9] B. Zhou and J. Pei, "Preserving privacy in social networks against neighborhood attacks," in *2008 IEEE 24th International Conference on Data Engineering*. IEEE, April 2008, pp. 506–515.

[10] 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*. New York, NY, USA: ACM, 2009, pp. 218–227.

[11] X. He, J. Vaidya, B. Shafiq, N. Adam, and V. Atluri, "Preserving privacy in social networks: A structure-aware approach," in *WI-IAT '09: Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, vol. 1. Washington, DC, USA: IEEE Computer Society, 2009, pp. 647–654.

[12] J. Lerner, *Role Assignments*, 2005, pp. 216–252.

[13] 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*. New York, NY, USA: ACM, 2009, pp. 1–10.

[14] M. Hay, G. Miklau, D. Jensen, D. Towsley, and P. Weis, "Resisting structural re-identification in anonymized social networks," *Proc. VLDB Endow.*, vol. 1, no. 1, pp. 102–114, 2008.

[15] E. Zheleva and L. Getoor, "Preserving the privacy of sensitive relationships in graph data," in *Privacy, Security, and Trust in KDD*, ser. Lecture Notes in Computer Science, F. Bonchi, E. Ferrari, B. Malin, and Y. Saygin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, vol. 4890, ch. 9, pp. 153–171.