Anonymizing Subsets of Social Networks with Degree Constrained Subgraphs

Sean Chester, Jared Gaertner, Ulrike Stege and S. Venkatesh
Department of Computer Science
University of Victoria, Victoria, Canada V8W 3P6
Email: schester@uvic.ca, jaredg@uvic.ca, stege@cs.uvic.ca, venkat@cs.uvic.ca

***Abstract*—In recent years, concerns of privacy have become more prominent for social networks. Anonymizing a graph meaningfully is a challenging problem, as the original graph properties must be preserved as well as possible. We introduce a generalization of the degree anonymization problem posed by Liu and Terzi. In this problem, our goal is to anonymize a given subset of nodes while adding the fewest possible number of edges. The main contribution of this paper is an efficient algorithm for this problem by exploring its connection with the degree-constrained subgraph problem. Our experimental results show that our algorithm performs very well on many instances of social network data.**

***Keywords*-privacy; social network; k-subset-anonymization; degree-constrained subgraphs;**

I. Introduction

This paper investigates the *k-subset-anonymity problem*. One is given a social network graph and a prespecified subset of members in the social network. The task is to ensure that within this subset, everyone is indistinguishable from at least $k-1$ others. In this way, the data can be published freely because an adversary would be unable to identify any member with certainty better than $\frac{1}{k}$.

What it means to be indistinguishable depends on the knowledge of the adversary and the context for the data. Several assumptions have been introduced in literature (see Section I-A for a review); a natural assumption is that for certain vertices their degree might be known. For example, a bipartite social network can be constructed from movies and reviewers where a link corresponds to a reviewer having reviewed a movie. However, it is extremely plausible that an adversary knows how many movies his target has reviewed. This is a case of *subset anonymity*, as there is no need to offer privacy to the movies.

We present the first algorithm for this more general formulation of k-anonymity. The algorithm is based on a novel reduction in the graph's complement to the degree-constrained subgraph problem, introduced by Gabow [1]. By focusing on subset anonymity, we greatly increase the chance of successfully anonymizing an input graph over the state-of-the-art technique for full graph anonymization of Liu and Terzi [2].

A. Related Work

The general idea of k-anonymity was introduced by [3]: records in tables could be made anonymous by suppressing data values until every record becomes identical to at least $k-1$ others with respect to the *quasi-identifiers*.[1] The problem of achieving k-anonymity was shown to be *NP*-Hard for $k \geq 3$ [4].[2]

k-Anonymity was recently adapted for social network graphs. [5] demonstrated that vertices of a graph can be uniquely identified by an adversary who possesses very reasonable background knowledge about the graph's structure, such as the degree of his target vertex. [2] introduced k-degree-anonymity as a means to protect against this sort of attack. They, and later [6] and [7], gave algorithms to k-degree-anonymize graphs. Subsequent to [2], numerous other notions of k-anonymity for graphs have been proposed [8]–[14], each assuming that the attacker has more sophisticated background knowledge about the structure of the graph, and then strengthening the privacy requirement for publishing.

[15] introduced the concept of subset anonymization, namely that one needs not anonymize the entire network. This is because different users have varied levels of privacy concern. Also, some vertices may not require privacy at all, such as the movies in our movie reviewer example. This concept was formalized for labeled graphs by [16], who also showed that achieving labeled subset anonymity is typically *NP*-Hard. However, for k-degree-subset-anonymity, the subject of this paper, the computational complexity is open.

Our work here differs from the above described k-anonymity papers in that we focus on subset anonymity. Other works that were designed specifically for anonymizing entire graphs do not directly apply to subset anonymization because they do not consider that certain edges have higher levels of desirability than others (Lemma 3.1). Furthermore, while [15] consider varied levels of concern for privacy and [16] formalize k-subset-anonymity, we are the first to tackle the algorithmic question of *how* to produce a good k-subset-anonymous graph.

Also important to this paper is the classic graph theory by [1], which offers a reduction from the degree-constrained subgraph problem to maximum matching; this can be solved with an established algorithm such as Edmonds' matching

[1] Quasi-identifiers are attributes such as postal code and birthdate which, when combined, can identify records uniquely.

[2] Subject to conditions on the size of the alphabet that were shown to be unnecessary in subsequent papers.

algorithm [17].

II. Background

We begin by describing the notations and definitions relevant to the *k-degree-subset-anonymity* problem.

A. Notation and Definitions

We assume familiarity with graph theory concepts such as a *graph* $\mathcal{G} = (\mathrm{V}, \mathrm{E})$, a *degree* d of a vertex $v \in \mathrm{V}$, and a *subgraph* of a graph. Assume all graphs are simple and undirected. A succinct (although not complete) representation of a graph (or subset of a graph) is its *degree sequence*, which is an ordered list of all its degrees:

Definition 2.1 (degree sequence of a set of vertices): The *degree sequence* $S_{\mathrm{X}} = \langle d_1, \ldots, d_{|\mathrm{X}|} \rangle$ of a set of vertices $\mathrm{X} \subseteq \mathrm{V}$ is the sequence of degrees for each $v \in \mathrm{X}$. We assume that the indices of vertices are assigned in decreasing order of degree so that the degree sequence is also sorted in decreasing order.

The primary objective in this paper is to produce k-degree-subset-anonymous graphs. Whether a subset of vertices is *k-degree-anonymous* is ascertainable from its degree sequence:

Definition 2.2 (k-anonymous degree sequence): A degree sequence S_{X} is said to be *k-anonymous* if every d_i appearing in S_{X} appears at least k times.

A set of vertices is said to be *k-degree-anonymous* if its degree sequence is k-anonymous:

Definition 2.3 (k-degree-anonymous set of vertices): A set X of vertices is said to be *k-degree-anonymous* if the degree sequence S_{X} of X is k-anonymous.

It is also useful to have a measure of how far a degree sequence is from being k-anonymous:

Definition 2.4 (unanonymity): The *unanonymity* of a set of vertices is the minimum number of degrees that must be removed from the set's degree sequence in order for it to become k-anonymous. If the unanonymity is zero, the set of vertices is k-degree-anonymous.

The first step in any k-degree-anonymity paper is to compute from the input graph's degree sequence a new target sequence. This gives rise to the important concept of a vertex's *deficiency*:

Definition 2.5 (degree deficiency of a vertex): Given two equal-length degree sequences, one termed the *source* ($S_{\mathrm{X}} = \langle d_1, \ldots, d_{|\mathrm{X}|} \rangle$) and the other termed the *target* ($S'_{\mathrm{X}} = \langle d'_1, \ldots, d'_{|\mathrm{X}|} \rangle$), the *deficiency* δ_i of a vertex $v_i \in \mathrm{X}$ is the difference between its degrees in the target and source sequences, $\delta_i = d'_i - d_i$.

B. Problem Definitions

We now introduce some known problems from graph theory that will be useful in Section III:

Problem 1. *Upper Degree-Constrained Subgraph (UDCS)* Given an input graph $\mathcal{G} = (\mathrm{V}, \mathrm{E})$ and an upper *degree constraint* $0 \leq c_i \leq d_i$ for each vertex v_i, find a maximum subgraph of $\mathcal{G}$ such for every vertex v_i, the degree is at most c_i.

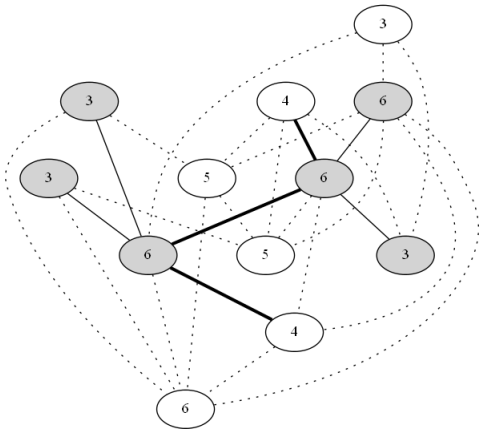


Figure 1. Example $\mathcal{G} = (\mathrm{V}, \mathrm{E})$ with dark vertices in X and lighter vertices in $\mathrm{V} \setminus \mathrm{X}$. The degrees are indicated within the vertices. The dotted edges are edges which are incident to a vertex in $\mathrm{V} \setminus \mathrm{X}$. The solid edges are eges which are incident to vertices in X. The bold edges are edges which have been added to $\mathcal{G}$ to make it 3-degree-anonymous.

Problem 2. *Degree-Constrained Subgraph (DCS)* Given an input graph $\mathcal{G} = (\mathrm{V}, \mathrm{E})$ and a *degree constraint* range $0 \leq l_i \leq u_i \leq d_i$ for each vertex v_i, find a maximum subgraph of $\mathcal{G}$ such for every vertex v_i, the degree is between l_i and u_i, inclusive.

We also introduce the goal problems in this paper, those of k-anonymity for graphs:

Problem 3. *k-degree-subset-anonymity (k-DSAP)* Given an input graph $\mathcal{G} = (\mathrm{V}, \mathrm{E})$ and an *anonymizing subset* $\mathrm{X} \subseteq \mathrm{V}$, produce an output graph $\mathcal{G}' = (\mathrm{V}, \mathrm{E} \cup \mathrm{E}')$ such that X is k-degree-anonymous and $|\mathrm{E}'|$ is minimized. In Figure 1 we illustrate k-DSAP.

We also introduce a relaxed version of the problem which is the one that we directly solve. However, our algorithm rarely (in fact, throughout our experimental evaluation in Section IV, never) produces a solution with an unanonymity higher than zero, and therefore effectively solves Problem 3.

Problem 4. *near-k-DSAP* Given an input graph $\mathcal{G} = (\mathrm{V}, \mathrm{E})$ and an *anonymizing subset* $\mathrm{X} \subseteq \mathrm{V}$, produce an output graph $\mathcal{G}' = (\mathrm{V}, \mathrm{E} \cup \mathrm{E}')$ such that *1*) the unanonymity of X is minimized and *2*) subject to 1), $|\mathrm{E}'|$ is minimized.

In the context of social networks, our problem corresponds to finding a minimum number of nonexistent relationships that can be added as noise distortion in order to ensure that nearly everyone has a 100% guarantee of being unrecognizable by degree beyond a $1/k$ probability.

III. Solving k-DSAP with DCS

In this section we describe how to compute a k-degree-subset-anonymous graph $\mathcal{G}'$ from an input graph $\mathcal{G}$. The algorithm hinges on reducing part of the k-DSAP problem to DCS, a reduction we describe in Section III-A. We then summarize the entire algorithm in Section III-B.

A. Realizing Target Degree Sequences by Means of DCS

Our objective is to demonstrate that DCS is a means with which to solve a crucial phase of the k-degree-subset-anonymity problem. In particular, given[3] the degree sequence of an optimal solution (the *target* degree sequence), we would have an immediate means by which to obtain the graph of the optimal solution. We begin by observing that certain edges are more valuable than others in terms of deriving an optimal solution. To arrive at the target degree sequence, the degree of every vertex must be increased by its deficiency. Edges that are strictly in X do double work in this regard, as shown in Lemma 3.1.

Lemma 3.1 (Edges in X*):* For any two arbitrary vertices $x_1, x_2 \in \mathrm{X}$ with deficiencies above zero and vertex $v \notin \mathrm{X}$, the edge (x_1, x_2) is doubly effective in comparison to either edge (x_1, v) or (x_2, v).

Proof: Every edge containing x_1 as an endpoint reduces the deficiency of x_1 by one. Likewise for x_2. Since the edge (x_1, x_2) contains both vertices as endpoints, it reduces the total deficiency in the graph by two at the cost of only one extra edge. On the other hand, both edges (x_1, v) and (x_2, v) only reduce the total deficiency by one. ■

Knowing that edges strictly in X are more desirable, we can conclude that solutions that have more of these are closer to optimal than solutions with less. For an output graph, denote the number of added edges strictly within X by $\mathrm{E}_{\mathrm{X,X}}$ and the number of added edges passing from X to $\mathrm{V} \setminus \mathrm{X}$ by $\mathrm{E}_{\mathrm{X,V}}$.

Corollary 3.2 (Large $|\mathrm{E}_{\mathrm{X,X}}|$ *is optimal):* For any two output graphs with the same degree sequence on X and the same induced subgraph on $\mathrm{V} \setminus \mathrm{X}$, the one with more edges in X is closer to an optimal solution.

Corollary 3.2 is the reason that DCS is so useful to us. We wish to identify as many edges strictly in X as possible in order to do as much work towards satisfying vertex deficiencies with as few edges as possible. In Lemma 3.3, we show that by cleverly selecting the set of edges for DCS to be only those that we especially want, we can maximize $\mathrm{E}_{\mathrm{X,X}}$.

Lemma 3.3 (DCS identifies maximum $|\mathrm{E}_{\mathrm{X,X}}|$*):* For a given graph $\mathcal{G} = (\mathrm{V}, \mathrm{E})$, vertices $\mathrm{X} \subseteq \mathrm{V}$, and ranges $[l_i, u_i]$ for each vertex $v_i \in \mathrm{X}$, the graph $\mathcal{G}'$ with maximum $|\mathrm{E}_{\mathrm{X,X}}|$ (and with each vertex degrees increased only by a value within its given range) is identified by the solution to DCS on the complement of the induced subgraph of $\mathcal{G}$ on X.

[3]In Section III-B we describe how to obtain the *target* degree sequence

Proof: By searching for a DCS solution in the induced subgraph of $\mathcal{G}$ on X, only edges strictly within X will be identified. Because the search is conducted on the complement, only edges that do not already exist in E will be identified. DCS maximizes the number of such edges. If there is a solution to the DCS, then the degree of every vertex is increased by a value within its given range. ■

By setting the set of edges for DCS to be only those highly desirable strictly-in-X-type edges and searching for a subgraph in the complement of $\mathcal{G}$ that is constrained to only find enough edges for each vertex to satisfy the deficiency, we can, indeed, find the optimal graph that corresponds to a target degree sequence.

Theorem 1 (Optimality of DCS): For a given graph $\mathcal{G} = (\mathrm{V}, \mathrm{E})$, vertices $\mathrm{X} \subseteq \mathrm{V}$, and target degree sequence $\mathcal{S}_T$, DCS identifies the optimal k-degree-subset anonymization of $\mathcal{G}$.

Proof: For each vertex $v_i \in \mathrm{X}$, let *reserve* be the number of vertices in $\mathrm{V} \setminus \mathrm{X}$ to which v_i is not connected. Then, construct a range $[l_i, u_i]$ such that u_i is the deficiency of v_i with respect to $\mathcal{S}_T$ and l_i is the larger of zero and $u_i - \textit{reserve}$. Then, by Lemma 3.3, we can identify the maximum number of edges that we can add strictly within X without overflowing the deficiency of any vertex and ensuring that any vertex with outstanding deficiency can be connected to sufficiently many *reserve* vertices. By Corollary 3.2, this is an optimal solution for the target degree sequence. If no DCS solution exists, then no k-degree-subset anonymization exists on $\mathcal{G}$ that achieves the target degree sequence. ■

A last point to note is that in Theorem 1 we established lower bounds to DCS to ensure that eventually those deficiencies not satisfied with edges strictly in X could be satisfied by vertices in $\mathrm{V} \setminus \mathrm{X}$. UDCS can work too, however, the risk is that it may produce some unanonymity.

B. The Complete Algorithm

In Section III-A we illustrated how, given a target degree sequence, we can produce the best solution graph using DCS. The main open question remaining is how to produce such a target degree sequence.

We follow the lead of [2], who offered an $\mathcal{O}(nk)$ dynamic programming algorithm to find the target degree sequence such that the sum of deficiencies is minimized over all vertices. That is to say, their dynamic programming algorithm, DP, finds the degree sequence which requires the fewest additional vertex endpoints in order to become k-degree-subset-anonymous.

Overall, the algorithm is described in Figure 2. We first find an optimal target degree sequence using the dynamic programming of [2]. We then identify the maximum number of strictly-in-X-type edges that can be added to the graph. Finally, we complete the anonymization by satisfying the outstanding deficiencies with edges from X to $\mathrm{V} \setminus \mathrm{X}$. In

k-DSAP_WITH_DCS
input: Graph $\mathcal{G} = (\mathrm{V}, \mathrm{E})$, $\mathrm{X} \subseteq \mathrm{V}$, integer k
$\mathcal{C} \leftarrow$ empty set of degree constraints
Compute degree sequence S_{X} of X
$S'_{\mathrm{X}} \leftarrow$ optimal k-anonymization of S_{X} with DP
foreach $v \in \mathrm{X}$
  outside $\leftarrow |\{u \in (\mathrm{V} \setminus \mathrm{X}) : \{u, v\} \notin \mathrm{E}\}|$
  required $\leftarrow \mathrm{deficiency}(v) - \mathrm{outside}$
  Add $[\max(0, \mathrm{required}), \mathrm{deficiency}(v)]$ to $\mathcal{C}$
$\mathcal{G}' \leftarrow$ complement of induced subgraph of $\mathcal{G}$ on X
$\mathcal{H} \leftarrow$ subgraph of $\mathcal{G}'$ satisfying degree constraints $\mathcal{C}$
foreach $e \in \mathcal{H}$
  $\mathrm{E} \leftarrow \mathrm{E} \cup \{e\}$
foreach $c_i = [c_l, c_u] \in \mathcal{C}$ with c_u unsatisfied in $\mathcal{H}$
  Add edges to E of form $\{v_i, u\}, u \in \mathrm{V} \setminus \mathrm{X}$ to satisfy c_u, whenever possible
return modified $\mathcal{G}$

Figure 2. Pseudocode description of the k-DSAP_WITH_DCS algorithm that transforms an input graph $\mathcal{G}$ into an output graph $\mathcal{G}'$ such that a prespecified subset of vertices X is k-degree-anonymous in $\mathcal{G}'$ by means of a reduction to the DCS problem.

Section IV, we demonstrate the efficacy of our algorithm by running it on a variety of datasets.

IV. Experimental Results

In this section we conduct an extensive empirical analysis of our algorithm by running it on three real world datasets. The intention is to observe for social-network-like inputs the success rate and efficiency of our algorithm. That is, how successfully do we minimize the distortion necessitated by k-anonymizing a social network and thus preserve the network's utility?

A. Experimental Setup

We implemented our algorithm in C++ using the Boost 1.47 libraries.[4] This included implementing the bipartite-substitute-based upper degree constraint satisfaction (UDCS) algorithm of [1], invoking the implementation of Edmonds' matching algorithm [17] provided by Boost.

We evaluated four dependent variables:

1) the success rate (produces a k-anonymous graph);
2) the time it takes to produce the output (execution time);
3) the number of edge additions within X;
4) the number of edge addtions from X to $\mathrm{V} \setminus \mathrm{X}$.

We measured the execution time of our algorithm with the *timer* class provided by Boost from the moment the input graph has been constructed until the moment our algorithm finishes reporting its solution.

The algorithm was run on the Wikipedia vote network, Enron email network and Epinion social network.[5] The independent variables of study are:

1) the anonymity requirement, $k \in \{2, 3, 4, 5\}$; and
2) the size of the anonymizing subset, $|\mathrm{X}| \in \{.2|\mathrm{V}|, .35|\mathrm{V}|, .5|\mathrm{V}|, .65|\mathrm{V}|, .8|\mathrm{V}|\}$.

We built the source in Visual Studio 2008 and then ran the experiments in a terminal on a 64-bit dual-core Intel®T8100 2.10GHz machine running Windows 7 with 4GB of memory.

B. Real World Dataset Experiment Results

The number of vertices of the graphs were 7,111 for the Wikipedia vote network, 36,692 for the Enron email network, and 75,879 for the Epinion social network. The number of edges of the graphs 103,689 for the Wikipedia vote network, 367,662 for the Enron email network, and 508,837 for the Epinion social network.

Success Rate. It was found that all experiments on all three of the real world graphs successfully anonymized a randomly chosen X.

Plots of Results. We report the running times for all experiments in Figure 3a. Each point in this plot corresponds to 10 trials (with the exception of the Epinions graph for $k = 5$ due to memory issues) and reports the average execution time. For all trials run, the longest any trial took was less than 700 seconds to anonymize the graph. Figure 3b shows the percentage of edges added, relative to the total number of edges in the graph, within X. Figure 3c shows the percentage of edges added, relative to the total number of edges in the graph, from X to $\mathrm{V} \setminus \mathrm{X}$.

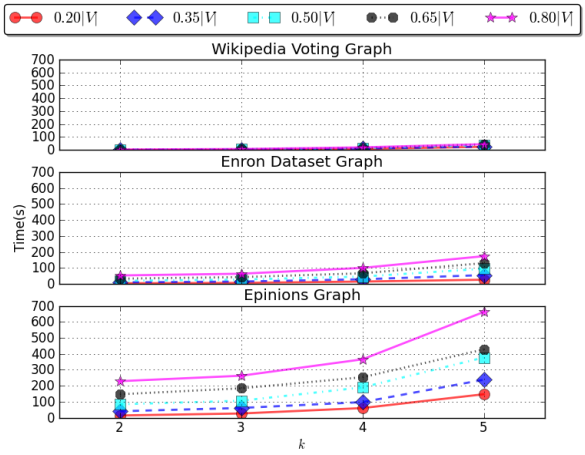
C. Discussion

In general, the results of the experiments were very encouraging. Below we present our interpretation of the results on unanonymity and efficiency.

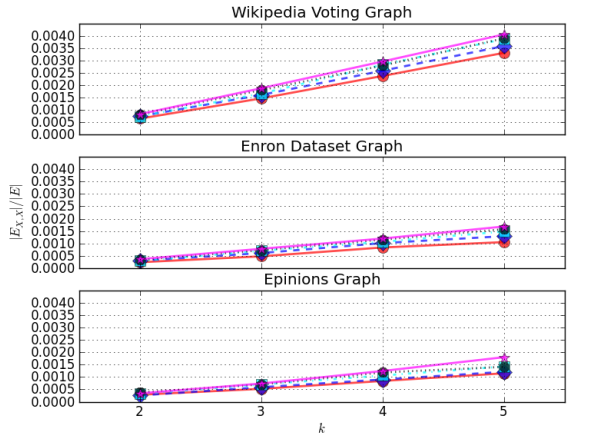
Unanonymity. Our first experiments were to evaluate the level of unanonymity left by using the UDCS version of the problem. As it turned out, across all trials on all datasets, the performance was perfect, even for the unusually high values ($.8|\mathrm{V}|$) of $|\mathrm{X}|$. This demonstrates that the UDCS version of the problem is an excellent alternative to DCS. As well, the percentage of edges added within X never exceeds 0.45% (Figure 3b) and the percentage of edges added from X to $\mathrm{V} \setminus \mathrm{X}$ never exceeds 1.8% (Figure 3c), thereby only altering a small percentage of $\mathcal{G}$ in order to make it k-degree-anonymous.

Execution Time. Social network anonymization is a task that only needs to be run once before data release, so execution time is not of utmost importance so long as it is reasonable. The execution time begins to increase as k increases. Still, it requires twelve minutes at worst and typically only a few minutes. In the context of anonymizing data once before publishing it, this is very acceptable.

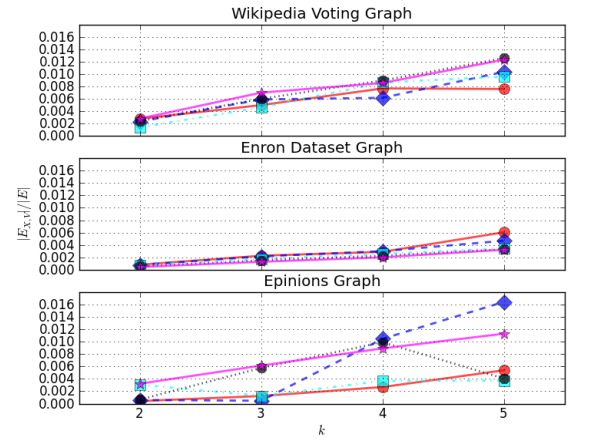
[4] http://www.boost.org
[5] Network data obtained from http://snap.stanford.edu/data/index.html



(a) Average time taken to run an experiment



(b) Average number of edges added within X relative to |E|



(c) Average number of edges added from X to V \ X relative to |E|

Figure 3. For each real world graph, the size of X is one of $\{0.2, 0.35, 0.5, 0.65, 0.8\}$ and the size of k is one of $\{2, 3, 4, 5\}$.

REFERENCES

[1] H. N. Gabow, "An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems," in *STOC*, 1983, pp. 448–456.

[2] K. Lui and E. Terzi, "Towards identity anonymization on graphs," in *SIGMOD*, 2008, pp. 93–106.

[3] L. Sweeney, "Achieving k-anonymity privacy protection using generalization and suppression," *Int. J. of Uncertain. Fuzziness and Knowl.-Based Syst.*, vol. 10, no. 5, pp. 571–588, 2002.

[4] A. Meyerson and R. Williams, "General k-anonymization is hard," in *PODS*, 2004.

[5] L. Backstrom, C. Dwork, and J. M. Kleinberg, "Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography," in *WWW*, 2007, pp. 181–190.

[6] S. Chester, B. M. Kapron, G. Ramesh, G. Srivastava, A. Thomo, and S. Venkatesh, "k-anonymization of social networks by vertex addition," in *ADBIS*, 2011, pp. 107–116.

[7] 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*, 2009, pp. 1–10.

[8] E. Zheleva and L. Getoor, "Preserving the privacy of sensitive relationships in graph data," in *PinKDD*, 2007, pp. 153–171.

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

[10] B. Zhou and J. Pei, "The k-anonymity and l-diversity approaches for privacy preservation in social networks against neighborhood attacks," *Knowl. and Inform. Syst.*, vol. 28, no. 1, pp. 47–77, 2011.

[11] B. Thompson and D. Yao, "The union-split algorithm and cluster-based anonymization of social networks," in *ASIACCS*, 2009, pp. 218–227.

[12] W. Wu, Y. Xiao, W. Wang, Z. He, and Z. Wang, "k-symmetry model for identity anonymization in social networks," in *EDBT*, 2010, pp. 111–122.

[13] G. Cormode, D. Srivastava, T. Yu, and Q. Zhang, "Anonymizing bipartite graph data using safe groupings," *The VLDB J.*, vol. 19, no. 1, pp. 115–139, 2010.

[14] S. Chester and G. Srivastava, "Social network privacy for attribute disclosure attacks," in *ASONAM*, 2011, pp. 445–449.

[15] M. Yuan, L. Chen, and P. S. Yu, "Personalized privacy protection in social networks," *PVLDB*, vol. 4, no. 2, pp. 141–150, 2010.

[16] S. Chester, B. M. Kapron, G. Srivastava, and S. Venkatesh, "Complexity of social network anonymization," *Soc. Netw. Anal. Min.*, March 2012.

[17] J. Edmonds, "Paths, trees, and flowers," *Canadian J. Math.*, vol. 17, pp. 449–467, 1965.