Privacy Risk In Graph Stream Publishing For Social Network Data

Nigel Medforth
School of Computing Science
Simon Fraser University
Burnaby, BC, Canada
nmedfort@cs.sfu.ca

Ke Wang
School of Computing Science
Simon Fraser University
Burnaby, BC, Canada
wangk@cs.sfu.ca

***Abstract*—To understand how social networks evolve over time, graphs representing the networks need to be published periodically or on-demand. The identity of the participants (nodes) must be anonymized to protect the privacy of the individuals and their relationships (edges) to the other members in the social network. We identify a new form of privacy attack, which we name the *degree-trail attack*. This attack re-identifies the nodes belonging to a target participant from a sequence of published graphs by comparing the degree of the nodes in the published graphs with the degree evolution of a target. The power of this attack is that the adversary can actively influence the degree of the target individual by interacting with the social network. We show that the adversary can succeed with a high probability even if published graphs are anonymized by strongest known privacy preserving techniques in the literature. Moreover, this success does not depend on the distinctiveness of the target nodes nor require the adversary to behave differently from a normal participant. One of our contributions is a formal method to assess the privacy risk of this type of attacks and empirically study the severity on real social network data.**

***Keywords*-social network; privacy; anonymity; data publishing;**

I. Introduction

Social networks are naturally represented by graphs: nodes correspond to participants within the network and edges (links) coincide with relationships between them. Participants of a social network typically want their sensitive information, including their relationships to the other individuals in the network, to remain private from the general public — but data miners and researchers want to analyze the raw data to discover interesting characteristics about particular social networks. Backstrom et al. showed that even if all of the identifying attributes of the participants were removed from the nodes, it was still possible for an adversary to re-identify the node of a participant by exploiting auxiliary information such as the degree of nodes [1]. A compromise is often reached between the data publishers and the data miners, resulting in both parties agreeing to some method that will be used to *anonymize* a snapshot of the live network data prior to its publication.

Motivation. The majority of privacy techniques for social network data focus on a single snapshot but one publication is not useful for mining evolving trends and patterns of social networks, such as how the popularity of individuals changes or how a disease spreads over time. To support such analysis, newer versions of social network data must be released periodically or on-demand. While multiple publications are extensively studied for relational data [5][14][13], those techniques only consider the participants' structured information (i.e., their identifiers, quasi-identifiers and sensitive attributes) and assumes that such information is immutable over time. In the case of social networks, the relationships between participants are typically in flux. Consequently, those techniques cannot be safely applied to evolving social networks.

In this paper, we identify a new class of privacy attack arising from publishing a sequence of graph data for evolving social networks. At any time i, an anonymized version G_i^* of the raw graph G_i is published, where G_i^* is produced by some graph anonymization method (e.g., degree k-anonymization [10], link perturbation [9], or k-isomorphism [6]). Prior to each publication, the adversary can actively influence the degree of a node in G_i by interacting with the social network. Such interactions may include simple friend requests in Facebook, posts on a message board, bids on Ebay, or email correspondences between different users. At time i, the adversary has access to $G_0^*, \cdots, G_i^*$ and tries to identify the candidate nodes for some targeted individual t by looking for those nodes that match with his knowledge about the degree of t in $G_0, \cdots, G_i$. The attack is successful if a small number of nodes match.

At first glance, the above attack sounds straightforward and elementary — given what is known about the privacy risks regarding multiple publications of relational data [5][14][13]. For example, with relational data, an adversary could use quasi-identifiers to identify the anonymity groups g_1 and g_2 for a targeted individual t in two different publications, and if the only common disease between g_1 and g_2 is `HIV`, then it is easy to infer that t has `HIV`. The key assumption here is the quasi-identifiers and sensitive attributes of an individual do not change over time. However, in the case of social networks we assume that only the graph is published and edges between nodes dynamically grow. Even if the adversary knows the exact degree of t in any G_i, every published graph G_i^* has been anonymized. For example, degree k-anonymization [10] ensures that at

least k nodes share the same degree, and link perturbation [9] randomly adds and deletes links between nodes. These operations alter the neighborhood structure of a node to make it difficult for the adversary to apply his knowledge. The adversary is further deterred by the power law of degree distribution where many nodes share similar degrees. Under these conditions re-identifying the node of a target individual is not straightforward.

Contributions. While the eventual goal is to invent methods for thwarting privacy attacks, this paper tries to answer several basic questions that are of broader importance: (1) how does the adversary re-identify the node of an individual from a sequence of anonymized graphs; (2) how to quantify the privacy risk of such re-identification; and (3) how severe and widespread are such attacks on real social networks. Our contributions are summarized as follows:

1) We identify a new type of privacy attack, which we named a *degree-trail attack*. We demonstrate that even if each published graph is anonymized by strong privacy preserving techniques, an adversary with little background knowledge can re-identify the node belonging to a known target individual by utilizing a sequence of published social network graphs. The adversary does not have to behave differently from a normal participant and the target node's degree can be similar to the degree of the other nodes in the graph.
2) We propose a new privacy technique, *stable link randomization*, which is based on link perturbation [9] but is better suited for multiple publications.
3) We present two alternative models to quantify the privacy risk of our degree-trail attack, assuming that the published graphs are anonymized by the stable link randomization.
4) We study the severity of the identified attack on real life social network data.

In the rest of the paper, Section II presents examples of degree-trail attacks; Section III formally defines the class of degree-trail attacks; Section IV presents two models of privacy risk of degree-trail attacks; Section V considers a more general class of degree-trail attacks that exploits the insertion time of nodes; Section VI presents an evaluation on the severity of the new attack using real life social network data. Section VII reviews related work and concludes the paper. A solution to preventing the identified attacks is very interesting but is beyond the scope of this paper.

II. Examples of Privacy Attacks

At each publication time i, there is a live graph G_i and the published graph G_i^*. G_i evolves into G_{i+1} after participants add or delete nodes or edges. A fixed and unique pseudonym called a *node ID* is associated with the node for each individual in every G_i and G_i^*, as assumed in [17][6]. A node ID is not related to the actual identity of the individual, but serves to track the nodes belonging to the same individual over multiple publications. This node ID serves three purposes: (1) It simplifies our discussion of the attack; (2) it is used to extract evolving trends in social network data, and (3) it results in a stronger privacy model by assuming that the adversary can locate the nodes representing the same individual in multiple publications.

At time i, prior to the publication of G_i^*, any participant (including an adversary) can create new nodes and add/delete links to other nodes in G_i. An adversary has access to $G_0^*, G_1^*, \cdots, G_i^*$ published up to time i. The adversary knows that some targeted individual is a participant of G_j and knows the degree of her target in G_j, for $j \leq i$. The adversary's goal is to identify the node in G_i^* that belongs to her target. Below, we illustrate how the adversary may achieve this goal even if all published graphs G_j^*, $j \leq i$, are anonymized by strong privacy preserving techniques. We consider two such techniques: *link perturbation* [8][15] and *k-isomorphism* [6].

Link perturbation. To protect the link information between two participants, Hay et al. outlined an algorithm for preventing link disclosure by introducing random noise to a graph [8]. This algorithm, link perturbation, produces G_i^* by randomly deleting m existing edges and randomly inserting m new edges to G_i. m is calculated using a fixed perturbation rate $m/|E|$, where $|E|$ is the number of existing edges in G_i. The perturbation rate is publicly known. [4] shows that link perturbation may achieve meaningful levels of identity obfuscation while still preserving characteristics of the original graph.

Consider the graphs G_0 and G_0^* in Figure 1. A circle represents a node and the letter inscribed within each node is its node ID. We label the target node by t but this label is not visible to the adversary; in fact, the adversary seeks to identify t. With the background knowledge on the target's degree 2 in G_0 and the perturbation rate of 10%, the adversary infers at some confidence level that the target's observed degree in G_0^* is expected to fall into the interval $[1, 3]$. By inspecting G_0^*, every node has an observed degree in this interval; therefore, every node in G_0^* is a candidate for t.

At time 1, the dashed circles in G_1 represent newly inserted nodes. To distance t's degree from other nodes, the adversary decides to increase her target's degree from 2 to 5 in G_1, by creating new connections to the target individual. Assuming that t gains 1 degree elsewhere, the adversary only needs to embed two more nodes, f and g, into G_1 and attach them to her target. Given the known perturbation rate, the adversary considers any node in G_1^* with an observed degree in the interval $[4, 6]$ to be a candidate for t. Upon inspecting G_1^*, the adversary would conclude that t and b are the candidates for her target t.

Prior to the next publication G_2^*, the adversary deletes both of her embedded nodes f and g from G_2, so t is left with a degree of 3 and the interval $[2, 4]$ for the observed

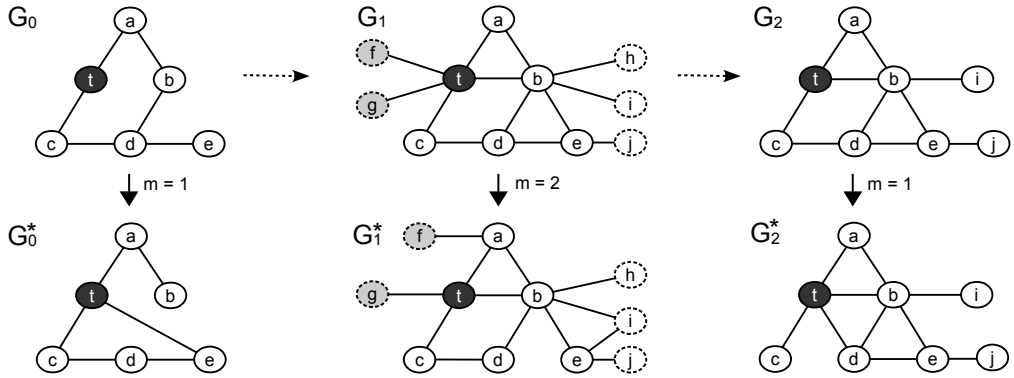


Figure 1. Attacks on randomized graphs

degree of t. Once obtaining G_2^*, the adversary discovers that $\{a, t, d, e\}$ have an observed degree in the interval $[2, 4]$. At this point, the adversary intersects this set with the previous candidate set $\{t, b\}$. Since the result contains only a single node, t, the adversary correctly identifies t as her target.

Remark 1. (1) The adversary cannot simply identify her embedded nodes as those having new node IDs because other participants are also creating new nodes. (2) Although the adversary can embed a more identifiable subgraph, doing so will increase her chance of being detected, thus, being rejected. To avoid that, a better strategy for the adversary is to use embedding that is similar to the normal growth of any node; thus, each single publication may return a large set of candidates. In this case, the success of the attack relies on observing several publications. (3) The only knowledge used by the adversary is the degree of the target node and the link perturbation algorithm used (including the perturbation rate).

k-isomorphism. *k-isomorphism* [6] produces G_i^* by transforming G_i into k isomorphic disconnected subgraphs, $g_1, \cdots, g_k$, through a minimum number of edge insertions and edge deletions. Consequently, for each edge (u, v) in any of these subgraphs, there will be $k-1$ "isomorphic links", one in each of the other $k-1$ subgraphs. This duplication creates uncertainty to associate links with target individuals. k-isomorphism implies k-automorphism [17], which implies degree k-anonymity [10].

k-isomorphism guards against adversaries with arbitrary neighborhood knowledge but is defenseless against background knowledge beyond that. Suppose that the node information $I(v)$ for each node v (such as Gender, Job, and Location) is published. To prevent re-identification of nodes using $I(v)$, the authors of [6] suggest detaching $I(v)$ from the node v by randomly partitioning the node set V into groups of size k and publishing $\{I(v_1), \ldots, I(v_k)\}$ for each group $\{v_1, \cdots, v_k\}$, therefore, breaking the linkage of the node information to each individual. Unfortunately, such grouping can defeat the purpose of k-isomorphism. For example, if the target individual is "male", any group $\{v_1, \cdots, v_k\}$ where $\{I(v_1), \ldots, I(v_k)\}$ contains no "males" can be excluded. This refinement creates a difference amongst the k-isomorphic subgraphs, thus, compromising the privacy provided by k-isomorphism.

In the case of multiple publications, the authors of [6] noted that the adversary could break k-isomorphism because the target node may belong to a different "k-isomorphic set" in different publications. To address this issue, they proposed to label all the k-isomorphic nodes with the *compound ID* formed by the set of node IDs of these nodes. For example, if u_1, u_2, u_3 are 3-isomorphic nodes from the 3-isomorphic subgraphs g_1, g_2, g_3, all of these nodes will be labeled by the compound ID $u_{\{1,2,3\}}$. Unfortunately, the use of compound IDs makes it impossible to track the nodes belonging to the same individual, which is essential for extracting evolving patterns of social networks.

III. DEGREE-TRAIL ATTACKS

In this section we formalize the class of attacks on a sequence of publications of social network graphs. We consider undirected, simple graphs. At time i, G_i denotes the raw graph and G_i^* denotes the published graph. Each node in G_i belongs to a participant and a link between two nodes represents a certain interaction between the two owning participants. $d(v)$ denotes the degree of a node v in G_i and $d^*(v)$ denotes the degree of a node v in G_i^*. G_i evolves into G_{i+1} as a result of adding and/or deleting nodes and edges. A fixed and unique node ID is associated with the node for each individual in every G_i and G_i^*. In section III-A, we present a technique called stable link randomization, which is better suited for multiple publications; then, in section III-B, we formalize our notion of a degree-trail attack, assuming that a published graph G_i^* is produced by stable link randomization.

A. Stable Link Randomization

Link perturbation [8] produces G_i^* from G_i by randomly inserting m edges and deleting m edges. For a single

publication, this ensures that an adversary cannot know with absolute certainty the degree of any node in G_i based on its degree in G_i^*. However, because the randomization of the same link is independent in each publication, whenever a link between the same pair of nodes is observed in *several* consecutive publications, there is a high probability that this edge actually exists in the raw graph. Additionally, if each G_i is randomized independently, the structural property (such as observed degree of a node) of G_i^* may be significantly different from that of G_{i+1}^* even if G_{i+1} is identical to G_i. This instability makes extraction of evolving patterns impossible. Finally, randomly inserting or deleting the same fixed number of m edges is not suitable for an evolving social network graph where the number of nodes and edges grows over time.

To address these issues, we propose *stable link randomization* for randomizing multiple versions of a graph. First, we fix the edge deletion rate ρ and edge insertion rate ϱ, instead of fixing the number of edges to be inserted and to be deleted. Therefore, an edge will be deleted with the probability ρ and kept with the probability $1-\rho$; a non-edge (i.e., an edge that does *not* exist in the graph) will be inserted with the probability ϱ and not inserted with the probability $1-\varrho$. Importantly, these probabilities remain constant as the graph grows or shrinks.

The key idea of stable link randomization is reusing the randomized edges from prior publications and only performing randomization on new edges/non-edges. The reuse of randomized edges ensures that each edge will be randomized only once. Specifically, for each edge/non-edge e in G_i, let e^* denote its randomized decision in G_i^*. If e also occurs in G_{i+1}, we reuse e^* in G_{i+1}^*. This is possible because the deletion rate ρ and insertion rate ϱ are same in both G_i^* and G_{i+1}^* and because the random trail of a link depends only on ρ, ϱ, and the link, not on any other links in the graph.

Let us describe this method in detail. We assume that G_{i+1} is obtained from G_i by a sequence of user-initiated *basic operations*: edge insertion is denoted by $+(u,v)$ and edge deletion is denoted by $-(u,v)$, where u and v are existing nodes in G_i, and node insertion is denoted by $+u$, where u is a new node. Deleting a node can be done by first deleting each of the edges adjacent to the node and then deleting the isolated node. For simplicity, we assume that isolated nodes are never deleted. After processing each basic operation on G_i, there is a corresponding G_i^*. Upon the request of the $(i+1)$-th publication, the most recent G_i^* is published as G_{i+1}^*. Below, we consider how to produce G_i^* after applying a single basic operation to G_i.

Processing $+(u,v)$: First, we insert the new edge (u,v) into G_i and G_i^*. Then, we need to perform a random deletion of the new edge (u,v) from G_i^*. Specifically, we toss a coin with ρ head probability and $1-\rho$ tail probability. If it lands heads, we delete (u,v) from G_i^*; if it lands tails, we keep (u,v) in G_i^*. Let us denote this operation on G_i^* by $+(u,v)^*$. Importantly, the new G_i^* inherits all edges from the old G_i^*. Note that (u,v) may already exist in G_i^* prior to performing $+(u,v)^*$ if it has been inserted by the randomization algorithm in a previous publication. In this case, $+(u,v)^*$ always overrides the current status of (u,v) in G_i^*.

Processing $-(u,v)$: First, we delete (u,v) from G_i. Since deleting (u,v) from G_i is equivalent to adding a new non-edge (u,v) into G_i, in G_i^* we need to perform a random insertion of (u,v) to account for the random insertion of this new non-edge. Specifically, we insert an edge (u,v) into G_i^* with ϱ probability and delete it with $1-\varrho$ probability. Let $-(u,v)^*$ denote this operation on G_i^*. Again, $-(u,v)^*$ always overrides the current status of (u,v) in G_i^*.

Processing $+u$: First, we insert the new node u into G_i and G_i^*. This means that every vertex $v \neq u$ in G_i gains a new non-edge (v,u). Therefore, we need to insert some number of edges over these new non-edges into G_i^* following the insertion rate ϱ. Let $+u^*$ denote the set of new edges (u,v) that are randomly selected with the insertion rate ϱ. We insert all the edges in $+u^*$ into G_i^*.

Remark 2. Besides the privacy reason mentioned above, stable link perturbation has two other benefits due to its inheritance of randomization from the previous published graph. (1) It is more efficient than a fresh randomization of all edges in G_i, especially for large and dynamic graphs. (2) It provides the stability on the randomized graph G_i^* in that G_i^* and G_{i+1}^* look similar if G_i and G_{i+1} look similar. This stability is crucial for data miners because a small change in the underlying graph should have little effect in the resulting publication.

B. Formalization of Degree-Trail Attacks

At time i, the adversary has access to all the graphs $G_0^*, \cdots, G_i^*$ published so far, generated by the stable link randomization method described above. The adversary also knows the edge deletion rate ρ and the edge insertion rate ϱ used by the stable link perturbation. In addition, the adversary knows that some target individual is a participant at all times j, for $0 \leq j \leq i$. The adversary's goal is to re-identify the node t in G_i^* that belongs to his targeted individual. To achieve this goal, the adversary actively attaches subgraphs to and/or removes subgraphs from the target node by creating connections to and/or removing connections from the target individual in the social network application. To avoid being caught or rejected, the adversary is limited to using "simple" subgraphs that are similar to what would be created by a normal participant. We consider the adversary who can create a small number of nodes and attach them to the target node or remove such nodes. This operation is called *embedding*. Without loss of generality, we assume that embedding operations are performed on G_j, $j > 0$, i.e., after observing G_0^*.

We assume that an adversary has the background knowledge about the target node t's degree in G_i, denoted $d(t)$. Such an assumption was made in previous works on a single publication of graphs [10][16][9][17][6]. Even armed with this knowledge, the graph randomization process described above may alter t's degree in the anonymized graph G_i^* and — even if it does not — there may be many nodes in G_i^* with the same degree as t. So, the adversary must identify the nodes v in G_i^* that are likely to be the node t, by taking into account the known degree $d(t)$, the observed degree of v in G_i^*, denoted $d^*(v)$, and the degree distortion due to the randomization process. This is done by estimating the probability that a node v with an observed degree $d^*(v)$ in G_i^* has the degree $d(t)$ in G_i. If this probability is high enough, v is considered as a candidate of t at time i. The *candidate set* of t at time i, denoted $C_i(t)$, is the set of all candidates of t in G_i^*.

In a single publication G_i^*, $C_i(t)$ may contain many candidates because many nodes in G_i^* have the same observed degree as t. For example, in a k-degree anonymized graph [10], if a node has some degree, at least k other nodes will have that degree. However, the adversary's power will be boosted by altering the degree $d(t)$ in a sequence of graphs G_j, $j \leq i$, and focusing on the nodes that are candidates for *all* of these G_j^*. This is because by definition, the target node t is expected to be a candidate in every G_j with a high probability. Thus the power of the adversary is boosted by refining the set of candidates to be those contained in the intersection $C_0(t) \cap \cdots \cap C_i(t)$.

Definition 1: A candidate of t *up to* time i is a node that is a candidate at time j for all $0 \leq j \leq i$. Thus the set of *candidates* of t up to time i is computed by

$$C^{(i)}(t) = \bigcap_{j=0}^{i} C_j(t) \tag{1}$$

$|C^{(i)}(t)|$ denotes the number of candidates in $C^{(i)}(t)$. □

Intuitively, $C^{(i)}(t)$ contains all the nodes that likely have a "similar" degree to that of t in every graph G_j, $j \leq i$, as inferred from the observed degree of nodes in G_j^*, the degree of t in G_j, and the randomization algorithm used for generating G_j^*. To the adversary, all these nodes are possible candidates of t; therefore, without further knowledge the probability of identifying the target node is $1/|C^{(i)}(t)|$. The adversary's goal is to reduce the size $|C^{(i)}(t)|$ by altering the degree of t in G_j by embedding nodes and attaching them to t, $j \leq i$. The exact nature of "similar" depends on the model for defining the candidate set $C_j(t)$. Below, we consider this issue.

IV. Two Models for Candidates

We present two models for defining the candidate set $C_i(t)$. At time i, the adversary has access to G_i^*, the degree $d(t)$ of the target node t in G_i, and the edge deletion rate ρ and edge insertion rate ϱ used by the randomization process to produce G_i^*. The randomization may (1) add zero or more edges to t, (2) remove zero or more edges from t, or (3) add or delete an edge between two non-target nodes. (3) does not affect the degree of the target node so we will only consider (1) and (2). Given these uncertainties, the adversary wants to identify the nodes that likely have a degree similar to $d(t)$ in G_i with a high probability. We propose two alternative models for quantifying this probability.

A. The Posterior Probability Model

The first model is based on the posterior probability that a node v has the same degree as the target node t in G_i, $d(t)$, given the node's observed degree $d^*(v)$ in G_i^*. If this probability is high enough, v is a candidate for t at time i, i.e., $v \in C_i(t)$. Without further knowledge, an adversary locates such candidates based on two things: $d(t)$ and $d^*(v)$. The best that the adversary can infer is the likelihood that a node v in G_i^* has the same degree as the target node in G_i, given the observed degree $d^*(v)$. This posterior probability and the notion of candidate sets based on it are defined below.

Definition 2 (Candidate set): Let $P_{\text{cand}}(d(t) \mid d^*(v))$ denote the probability that a node v has the same degree as the target in G_i, i.e., $d(t)$, given its observed degree $d^*(v)$. Given a threshold λ, the candidate set of t at time i, $C_i(t)$, contains all nodes v in G_i^* such that $P_{\text{cand}}(d(t) \mid d^*(v)) > \lambda$. □

$P_{\text{cand}}(d(t) \mid d^*(v))$ is also the probability that t has gained $d^*(v) - d(t)$ edges if $d^*(v) \geq d(t)$ or has lost $d(t) - d^*(v)$ edges if $d^*(v) < d(t)$ due to the stable link randomization. Therefore, this probability is closely related to answering the following question: given ρ and ϱ, how probable is it that some number of edges, say r, could be inserted into or deleted from the target node due to the stable link randomization? We refer to those probabilities as $P_{ins}(r)$ and $P_{del}(r)$, respectively. Insertions and deletions could happen simultaneously but since they are chosen from two different disjoint sets of edges, $P_{ins}(r)$ and $P_{del}(r)$ are independent and can be considered separately.

Let us determine $P_{ins}(r)$ and $P_{del}(r)$. The number of non-edges associated with the target node, $n(t)$, in a graph of N nodes is $n(t) = N - d(t) - 1$. Each of these non-edges have ϱ probability to be inserted. Therefore the probability that exactly r edges could be added to the target node t is equal to:

$$P_{ins}(r) = \binom{n(t)}{r} \varrho^r (1-\varrho)^{n(t)-r} \tag{2}$$

Similarly, the probability of deleting exactly r of t's edges in G_i^* is:

$$P_{del}(r) = \binom{d(t)}{r} \rho^r (1-\rho)^{d(t)-r} \tag{3}$$

To compute $P_{\text{cand}}(d(t) \mid d^*(v))$, consider two cases below. Let $\Delta(v) = |d(t) - d^*(v)|$.

$$P_{\text{cand}}(d(t) \mid d^*(v)) = \begin{cases} P_{\text{ins}}(d(t) \mid d^*(v)) & \text{if } d^*(v) \geq d(t) \\ P_{\text{del}}(d(t) \mid d^*(v)) & \text{otherwise} \end{cases} \quad (4)$$

$P_{\text{ins}}(d(t) \mid d^*(v))$ is the probability that the graph randomization has increased the degree of t by $\Delta(v)$, and $P_{\text{del}}(d(t) \mid d^*(v))$ is the probability that the randomization has decreased the degree of t by $\Delta(v)$. To calculate $P_{\text{ins}}(d(t) \mid d^*(v))$, we must consider all possible situations that could have allowed the observed degree to increase by $\Delta(v)$: if the stable link randomization added $\Delta(v)+1$ edges to it, then it must have deleted 1 edge from it as well in order for that node to have a degree of $d^*(v)$; if it added $\Delta(v)+2$ edges to it, then it must have deleted 2 edges from it, and so on. Since stable link randomization will only delete existing edges, no more than $d(t)$ edges could ever be removed from the target node. We have:

$$\begin{aligned} P_{\text{ins}}(d(t) \mid d^*(v)) &= \sum_{r=0}^{d(t)} [P_{ins}(\Delta(v)+r) P_{del}(r)] \\ &= \sum_{r=0}^{d(t)} [P_{ins}(d^*(v) - d(t) + r) P_{del}(r)] \\ P_{\text{del}}(d(t) \mid d^*(v) &= \sum_{r=0}^{d^*(v))} [P_{del}(\Delta(v)+r) P_{ins}(r)] \\ &= \sum_{r=0}^{d^*(v)} [P_{del}(d(t) - d^*(v) + r) P_{ins}(r)] \end{aligned}$$

These computations only need $d(t)$, $d^*(v)$, ρ, and ϱ, which are all known to the adversary. Since $d(t)$ and $d^*(v)$ are not very large, these probabilities can be computed efficiently.

B. The Confidence Interval Model

The second model for defining $C_i(t)$ is based on some confidence interval for the observed degree of t. Given the edge deletion rate ρ and the edge insertion rate ϱ of the graph randomization, and the known degree $d(t)$ of the target node t, we can derive the expected value of the observed degree of t and some confidence interval covering the expected value. One complication is that edge deletions and edge insertions follow different parameters, i.e., ρ and ϱ. Below, we model the effect of these operations by a sequence of Poisson trials.

Let $D_1, \cdots, D_{d(t)}$ denote the *deletion variables* for the $d(t)$ existing edges adjacent to t. $D_i = 0$ represents that the i-th edge adjacent to t is deleted and $D_i = 1$ represents that the i-th existing edge adjacent to t is kept. Following our randomization algorithm, $Pr[D_i = 1] = 1-\rho$ and $Pr[D_i = 0] = \rho$. Let $I_1, \cdots, I_q$ denote the *insertion variables* for the $q = |V| - d(t) - 1$ non-edges of t. $I_i = 0$ represents that the i-th non-edge of t is not inserted and $I_i = 1$ represents that the i-th non-edge of t is inserted. $Pr[I_i = 1] = \varrho$ and $Pr[I_i = 0] = 1-\varrho$. The observed degree of t in G_i^* is equal to $S = \sum_i D_i + \sum_i I_i$. Since D_i and I_i are variables for independent Poisson trials, the expected value of S, $E(S)$, is equal to the sum of the expected values of its component variables:

$$E(S) = d(t) * (1-\rho) + q * \varrho \quad (5)$$

According to Chernoff bound [7], for a relative error bound $\delta > 0$, we have

$$Pr[|S - E(S)| \geq \delta E(S)] \leq 2exp[-E(S)\delta^2/4] \quad (6)$$

Let

$$\theta = 1 - 2exp[-E(S)\delta^2/4] \quad (7)$$

$$\delta = \sqrt{-4\frac{ln\frac{(1-\theta)}{2}}{E(S)}} \quad (8)$$

The above Chernoff bound is rewritten into

$$Pr[|S - E(S)| \leq \delta E(S)] \geq \theta \quad (9)$$

Intuitively, Equation (9) says that if G_i is randomized many times, in at least θ percent of the cases the observed degree of t will fall into the interval $[E(S)-\delta E(S), E(S)+\delta E(S)]$. δ is the *relative error*, θ is the *confidence level*, and $[E(S)-\delta E(S), E(S)+\delta E(S)]$ is the *confidence interval*. Note that δ and θ are not independent, in fact, they are related by Equations (7) and (8).

Definition 3 (Candidate set): For a given confidence level θ in $(0,1)$, $C_i(t)$ contains all nodes in G_i^* whose observed degree falls into the interval $[E(S) - \delta E(S), E(S)+\delta E(S)]$, where $E(S)$ is the mean of observed degree of t defined by Equation (5) and δ is defined by Equation (8). □

For a given confidence level θ, the adversary can compute the confidence interval because $d(t)$, ρ and ϱ are known. The next corollary follows from the above discussion.

Corollary 4.1: Given a confidence level θ in $(0,1)$, the probability, over all randomized graphs G_i^*, that $C_i(t)$ contains the target node t is at least θ, where θ is defined in Equation (7). □

From Equation (7), a larger error bound δ increases the probability that $C_i(t)$ contains the target node t, but also leads to a larger $C_i(t)$, which reduces the power of attacks. The adversary has to balance the two.

V. Time-Refined Attacks

The basic attack in Section IV ignores what the adversary can learn about the insertion time of the nodes the target must be adjacent to. In this section, we examine how the insertion time of nodes can be used for attack.

At time i, the adversary influences the degree of the target node t by inserting new nodes and attaching them to t as well as attaching existing nodes to t. Every new node

inserted into G_i is assigned a *new* node ID. Consequently, any nodes in G_i^* having a node ID that was not observed in any previous publications can be marked as being inserted at time i. For this reason, the insertion time of every node is always known to the adversary. The next example illustrates how the adversary can use this knowledge to refine the basic degree-trail attack. For simplicity, we temporarily ignore randomization in the following example.

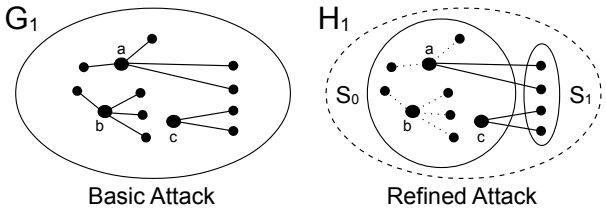


Figure 2. Basic vs. Refined Attack

In Figure 2, suppose that the adversary knows that the target node t has degree 4 in G_1, two of which come from two nodes embedded by him. With the notion of degree, both a and b are probable candidates of t and the adversary cannot differentiate between them. In the graph H_1, nodes are partitioned according to their insertion time, where S_i contains the nodes with insertion time i, $i = 0, 1$.

Since the adversary embedded two nodes to the target node in G_1, the target node must be adjacent to at least two nodes in S_1. This knowledge immediately allows the adversary to eliminate b as an option as it is not adjacent to any node in S_1, leaving the adversary to conclude that a is the target node. So the next question is how could the adversary uses this observation in light of graph randomization? To answer this, we must first refine the notion of degree by insertion time.

Definition 4: Consider G_i. For $0 \leq j \leq i$, let V_j be the set of nodes inserted at time j, called *j-nodes*, and let S_j be the set of j-nodes in V_j that remain in G_i. The *j-degree* of a node v in G_i refers to the number of edges adjacent to the nodes in S_j. $d_j(v)$ denotes the j-degree of v in G_i and $d_j^*(v)$ denotes the observed j-degree of v in G_i^*. □

At time i, for $0 \leq j \leq i$ the adversary can influence $d_j(t)$ by either inserting a new edge between t and a j-node or deleting an existing edge between t and a j-node. We assume that the adversary knows the j-degree of the target node t, $d_j(t)$, which is a more elaborated form of the degree knowledge considered in the previous section. Below, we extend the notion of candidates to the case for time-refined attacks.

Definition 5: Given G_i^*, for $1 \leq j \leq i$, we define a bipartite graph $H_j^*(C^{(i-1)}(t), S_j, E_j^*)$, where E_j^* contains the edges in G_i^* between the nodes in $C^{(i-1)}(t)$ and the nodes in S_j. □

H_j^* is the subgraph of G_i^* involving only the edges between the candidates in $C^{(i-1)}(t)$ and the nodes that were inserted at time j. Assuming that $C^{(i-1)}(t)$ has been computed at time $i-1$, the adversary can construct H_j^* because S_j is available. With H_j^*, the adversary can tell the observed j-degree of each candidate in $C^{(i-1)}(t)$. If a candidate v in $C^{(i-1)}(t)$ satisfies the adversary's expectation on the observed j-degree of t in G_i^*, for all $0 \leq j \leq i$, v is a candidate in $C^{(i)}(t)$. Based on this idea, we consider the two models separately.

In the posterior probability model, $P_{\text{cand}}(d_j(t) \mid d_j^*(v))$ measures the probability that a node v has the j-degree $d_j(t)$ in G_i, given the observed j-degree $d_j^*(v)$ in G_i^*. Since this probability depends only on $d_j^*(v)$ and $d_j(t)$, $P_{\text{cand}}(d_j(t) \mid d_j^*(v))$ can be computed by applying the method described in Section IV-A to the subgraph H_j^*.

Definition 6 (The posterior probability model): For a given bound λ, the candidate set for t up to time i, $C^{(i)}(t)$, is the set of the nodes v in $C^{(i-1)}(t)$ such that $P_{\text{cand}}(d_j(t) \mid d_j^*(v)) > \lambda$ for $0 \leq j \leq i$. □

In the confidence interval model, the expected observed j-degree of t in G_i^*, denoted $E_j(S)$, is computed by

$$E_j(S) = d_j(t) * (1 - \rho) + q_j * \varrho \tag{10}$$

where q_j is the number of non-edges from t to j-nodes in G_i: $q_j = |S_j| - d_j(t) - 1$ if $j = 0$ (recall that t has insertion time 0) and $q_j = |S_j| - d_j(t)$ if $j > 0$.

Definition 7 (The confidence interval model): The candidate set of t up to time i, $C^{(i)}(t)$, is the set of the nodes v in $C^{(i-1)}(t)$ such that, for $0 \leq j \leq i$, $d_j^*(v)$ falls into the interval $[E_j(S) - \delta E_j(S), E_j(S) + \delta E_j(S)]$, where $E_j(S)$ is given by Equation (10). □

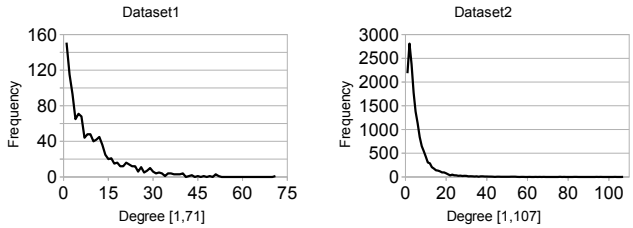
VI. Experiment

In this section we study the extent to which degree-trail attacks succeed on real social network graphs that were anonymized using stable link randomization (Section III-A).

A. Methodologies

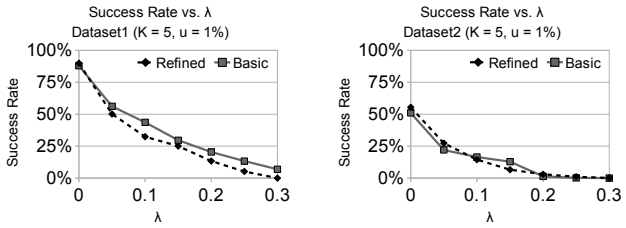
We considered two real life data sets, whose degree frequency are shown in Figure 3. Dataset 1 is the e-mail network of *University Rovirai Virgili* (URV) with 1,132 nodes and 5,450 edges [12]. Dataset 2 is the *Newman*'s scientific collaboration network [11] with 16,264 nodes and 47,594 edges. We started with these graphs as G_0. First, we selected a node at random from G_0 as the target node t. Then, we generated each subsequent graph G_i from $G_{i-1}(V, E)$ to simulate the growth of nodes and edges. This simulation uses three growth parameters c, s and u. Initially, let G_i be G_{i-1}. We augment G_i in three steps:

1) randomly select a subset S of nodes of size $c|V|$ from G_i and add the target node t to S if it is not there.
2) add $s|V|$ new nodes to both G_i and S.
3) randomly add $u|N(S)|$ new edges between the nodes in S to G_i, where $|N(S)|$ is the number of non-edges between nodes in S.



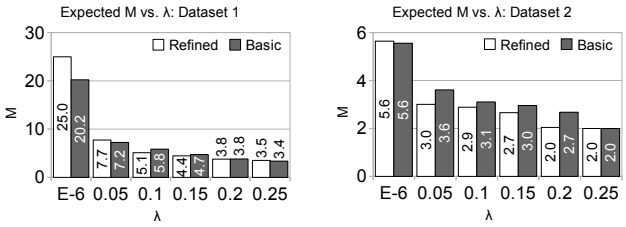
(a) Dataset 1 (b) Dataset 2

Figure 3. Degree Distribution



(a) Dataset 1 (b) Dataset 2

Figure 4. Success rate vs. λ



(a) Dataset 1 (b) Dataset 2

Figure 5. Expected # of publications for first success (M) vs. λ

Intuitively, S models a subset of nodes that are dynamic in the growth of their edges. S contains some existing nodes, some new nodes, and always contains the target node t. By randomly adding new edges between the nodes in S, the embedding to t is no different from the growth of other nodes in S. This ensures that the adversary does not behave differently from other nodes in S. The parameter c controls the "focus" on t, and the parameters s and u control the growth in the number of nodes and the number of edges. A small c and a large s or u will increase the probability that t will gain new edges in G_i. Unless otherwise specified, we fix c at 10%, s at 1%, and u at 1%.

G_i^* is produced from G_i using the stable link randomization — with the following stipulation: since G_0 is the first graph in the sequence of raw graphs, the creation of G_0^* is identical to the traditional link perturbation [9] on G_0 except for the fact we fix the edge deletion rate ρ and edge insertion rate ϱ. Based on Hay et al.'s findings on link perturbation [8], we fix ρ at 10%. ϱ is calculated based on the notion of deleting m existing edges from G_0 and inserting m new edges into G_0. In other words, ϱ is set at $\rho|E(G_0)|/|N(G_0)|$, i.e., $\approx 0.859\%$ and $\approx 0.036\%$ for Datasets 1 and 2 respectively, where $|E(G_0)|$ and $|N(G_0)|$ denote the number of edges and the number of non-edges in G_0.

For a given privacy degree k, an attack *converges* at time i if $C^{(i)}(t)$ contains at least one but no more than $k-1$ candidates. An attack *succeeds* at time i if it converges at time i and $C^{(i)}(t)$ actually contains the target node t. Unless otherwise specified, we fix k at 5. Note that the adversary knows when an attack converges but cannot tell whether an attack succeeded as he cannot verify whether t actually belongs to $C^{(i)}(t)$. The notion of "success" is only for our own evaluation. To reduce the bias of results, we generated 10,000 distinct graph sequences $G_0, \cdots, G_i$ (and the corresponding published sequence $G_0^*, \cdots, G_i^*$) for each data set. t was randomly chosen from G_0 and fixed throughout a particular graph sequence. The *success rate* is defined as the percentage of the cases in which an attack succeeds, and the *convergence rate* is defined as the percentage of the cases in which an attack converges. Below, we examine success rate and convergence rate separately.

B. Success Rate

In the posterior probability model in Section IV-A, we use the condition $P_{\text{cand}}(d(t) \mid d^*(v)) > \lambda$ to tell if a node v is a candidate in $C_i(t)$. We asked ourselves the following questions: (1) how does λ affect the success rate of the attack, and (2) how many publications would be required on average for an attack to succeed for a given λ? To answer these questions, we ran two experiments to assess the success rate wrt λ for the basic degree-trail attack and the time-refined attack. Figure 4 reports the success rate and Figure 5 reports the minimum number of publications required for a success.

As λ decreases, the success rate increases, assuming an unlimited number of publications. A smaller λ affects the success rate in two ways: (1) with a smaller λ it is less likely that the target node t will be wrongly discarded from $C_i(t)$. (2) with a smaller λ, $C_i(t)$ tends to be larger and an adversary would require more publications for an attack to succeed, as shown in Figure 5. Initially, we were concerned that setting λ too low would result in each $C_i(t)$ being too large to converge quickly on a small set of nodes. However, even when we set λ at 0.000001, we discovered that $C^{(0)}(t)$ only contained $\approx 61\%$ and $\approx 78\%$ of the nodes from Dataset 1 and Dataset 2 respectively. The percentage of nodes in each subsequent $C^{(i)}(t)$ dropped sharply with each publication. These results suggest that a "conservative" approach of using a small threshold λ but examining more publications presents a more effective attack.

Surprisingly, while the success rate of the posterior probability model was fairly high in Dataset 1, the success rate in Dataset 2 was far less promising. Upon investigation, it

appears that many nodes in Dataset 2 share a similar low degree (Figure 3(b)), consequently, it is harder to converge to a small candidate set that contains the target node.

Interestingly, the basic degree-trail attack has a similar success rate compared to the time-refined attack — a finding that did not match our expectation that the refined attack was more powerful. A closer look reveals that the refined attack model suffered a larger estimation error. Specifically, this attack refines the degree of a node into j-degrees according to the insertion time j of adjacent nodes and estimates the posterior probability based on j-degree. With each j-degree being small, such estimation deteriorates in accuracy. This is similar to tossing a coin only a few times, which does not give an accurate estimation of head/tail probability.

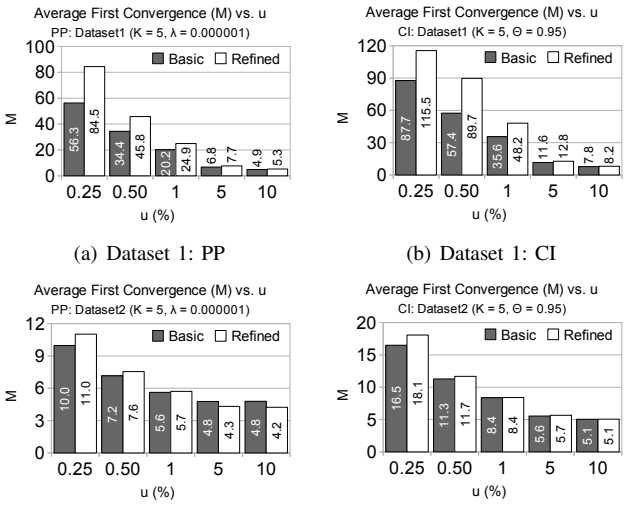
C. Convergence Rate

The remaining experiments focused on the number of publications required for convergence of attacks and the differences between the posterior probability model (PP) and the confidence interval model (CI). The probability threshold λ for PP is set to 0.000001 and the confidence level θ for CI is set to 95%.

Figure 6 shows that as the edge growth rate u increases, the average number of publications (M) required for convergence decreases. The reason for this was because as the amount of activity increased within the subset S prior to each publication, it became more likely for the target and non-target nodes to diverge in terms of degree similarity. This implies that the more active a social network is, the more likely an adversary will be able to locate her target without being detected. The number of publication required for convergence is much smaller for the larger Dataset 2 because the accuracy of both PP and IC improves as the number of random trials performed increases.

Figure 7 shows the number of publications (M) required for convergence vs different settings of privacy degree k. M did not vary considerably for values of k between 3 and 10 for each individual attack. This is not surprising since many of the publication sequences converged when the candidate set reduced from ≥ 10 nodes in G^*_{i-1} to < 3 nodes in the final publication G^*_i. On the smaller Dataset 1, the refined attack requires more publications to converge than the basic attack because the larger estimation error of head/tail probability of each j-degree affects more a small graph. This result further suggests that the refined attack is not suitable for small graphs. Figure 8 illustrates the convergence rate *within* a certain number of publications (M). For example, a point (6, 25%) means that in 25% of cases the attack is able to converge in ≤ 6 publications.

These empirical studies suggest two main findings: First, a degree-trail attack succeeds frequently on real social network graphs that have been anonymized by powerful link randomization techniques. Second, an adversary may not require many publications in order to successfully attack a target



(a) Dataset 1: PP (b) Dataset 1: CI

(c) Dataset 2: PP (d) Dataset 2: CI

Figure 6. Average M vs. u (k = 5)

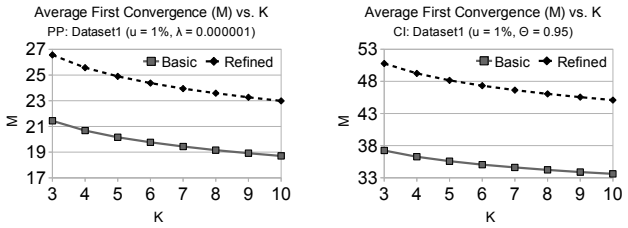
within a large active social network. These findings highlight the vulnerability of privacy breaches when publishing social network graph data. We feel that our results show that there is a need for more research in this area.

VII. Related Work and Conclusion

Backstrom et al. were the first to propose an active attack on social networks [1]. The primary disadvantage of their attack was that no isomorphic subgraphs could exist within the live graph. Furthermore, their attack did not consider published data that was anonymized beyond the removal of identifying and quasi-identifying attributes and did not consider multiple publications. Passive attacks have been extensively studied in the literature where the adversary attacks an already published network [2][8][10][16][17] for identity disclosure, and [6][8] for link disclosure. All those works consider a single snapshot of a static social network.

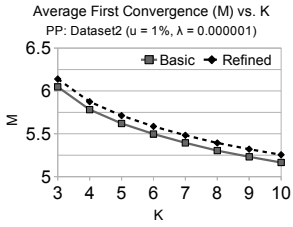
There is little work on preventing privacy attacks in the context of publishing dynamic social network data. One such work is [3] where the authors propose to reduce the privacy loss by predicting the growth of links and factoring such growth in a group-based anonymization approach. They hide the mapping between a node and the corresponding entity called label by partitioning the set of nodes into groups of size k and assigning the label list for the group to each node in the group. However, the published graph is still vulnerable to an adversary with the auxiliary information on neighborhood structure (such as degree). In addition, with all nodes in a group being associated with the same label list, the correspondence between the nodes in different publications is lost, rendering mining sequential patterns impossible.

Our work considered an adversary armed with auxiliary information about the degree of the target and the ability to

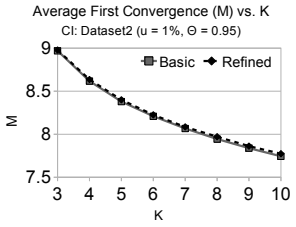


(a) Dataset 1: PP

(b) Dataset 1: CI

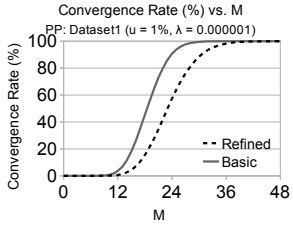


(c) Dataset 2: PP

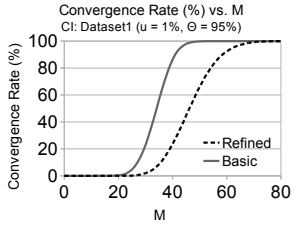


(d) Dataset 2: CI

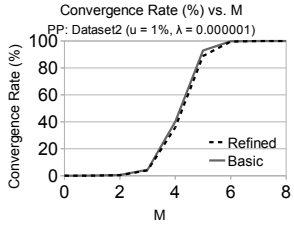
Figure 7. Average M vs. k (u = 1%)



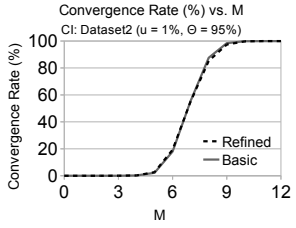
(a) Dataset 1: PP



(b) Dataset 1: CI



(c) Dataset 2: PP



(d) Dataset 2: CI

Figure 8. Convergence Rate within M publications (u = 1%; k = 5)

actively influence such degrees and coordinate the attacks over multiple publications. The contribution of this work is a preliminary investigation on several basic questions of broader importance: what does it mean to compromise privacy, what background information does the adversary need, what behaviors of the adversary are required, where does the power of the attack come from, and how widespread are such attacks on real life data. We hope these findings are useful for future work in this area.

ACKNOWLEDGMENT

Ke Wang's work is supported by a Discovery Grant from the Natural Sciences and Engineering Research Council of Canada. -

REFERENCES

[1] Lars Backstrom, Cynthia Dwork, and Jon Klienberg. Wherefore art thou r3579x? anonymized social networks, hidden patterns, and structural steganography. In *WWW*, 2007.

[2] K. Bai, Y. Liu, and P. Liu. Prevent identity disclosure in social network data study. In *ACM CCS*, 2009.

[3] Smriti Bhagat, Graham Cormode, Balachander Krishnamurthy, and Divesh Srivastava. Privacy in dynamic social networks. In *WWW*, 2010.

[4] Francesco Bonchi, Aristides Gionis, and Tamir Tassa. Identity obfuscation in graphs through the information theoretic lens. In *ICDE*, 2011.

[5] J.-W. Byun, Y. Sohn, E. Bertino, and N Li. Secure anonymization for incremental datasets. In *SDM Workshop*, 2006.

[6] Jams Cheng, Ada Wai-Chee Fu, and Jia Liu. K-isomorphism: Privacy preserving network publication against structural attacks. In *SIGMOD*, 2010.

[7] Herman Chernoff. A note on an inequality involving the normal distribution. *Ann. Probab*, 9(3):533–535, 1981.

[8] Michael Hay, Gerome Miklau, David Jensen, Philipp Weis, and Siddharth Srivastava. Anonymizing social networks. Technical report, SCIENCE, 2007.

[9] Micheal Hay, Gerome Miklau, David Jensen, Don Towsley, and Philipp Weis. Resisting structural re-identification in anonymized social networks. In *PVLDB*, 2008.

[10] Kun Liu and Evimaria Terizi. Towards identity anonymization on graphs. In *SIGMOD*, 2008.

[11] Newman M. E. J. The structure of scientific collaboration networks. In *The National Academy of Sciences of the USA*, volume 98, pages 404–409, 2001.

[12] A. Diaz-Guilera F. Giralt R. Guimera, L. Danon and A. Arenas. Self-similar community structure in a network of human interactions. In *Physical Review E*, volume 68, 2003.

[13] K. Wang and B. Fung. Anonymizing sequential releases. In *SIGKDD*, 2006.

[14] X. Xiao and Y. Tao. m-invariance: towards privacy preserving re-publication of dynamic datasets. In *SIGMOD*, 2007.

[15] Xiaowei Ying and Xintao Wu. Randomizing social networks: a spectrum preserving approach. In *SDM*, 2008.

[16] B. Zhou and J. Pei. Preserving privacy in social networks against neighborhood attacks. In *ICDE*, 2008.

[17] Lei Zou, Chen Lei, and M. Oszu Tamer. K-automorphism: A general framework for privacy preserving network publication. In *VLDB*, 2009.