A Differentially Private Estimator for the Stochastic Kronecker Graph Model

Darakhshan Mir
Rutgers University
mir@cs.rutgers.edu

Rebecca N. Wright
Rutgers University
rebecca.wright@rutgers.edu

ABSTRACT

We consider the problem of making graph databases such as social networks available to researchers for knowledge discovery while providing privacy to the participating entities. We use a parametric graph model, the stochastic Kronecker graph model, to model the observed graph and construct an estimator of the "true parameter" in a way that both satisfies the rigorous requirements of differential privacy and demonstrates experimental utility on several important graph statistics. The estimator, which may then be published, defines a probability distribution on graphs. Sampling such a distribution yields a synthetic graph that mimics important properties of the original sensitive graph and consequently, could be useful for knowledge discovery.

Categories and Subject Descriptors

K.4.1 [**Computers and Society**]: Public Policy Issues—*privacy*; H.1.1 [**Models and Principles**]: Systems and Information Theory

General Terms

Security

Keywords

Privacy, Information Theory

1. INTRODUCTION

As graph databases such as social networks become ubiquitous, researchers have an unprecedented opportunity to understand and analyze complex social phenomena. For example, access to a social network may help researchers track the spread of an epidemic or a sexually-transmitted disease in a community. While society would like to encourage such scientific endeavors, if individuals run the risk of being identified, they may be apprehensive of participating in, or making their social network data available for, such studies. To ensure that public policy promotes such scientific projects, we are faced with the problem of providing researchers with a fairly accurate picture of the quantities or trends they are looking for without disclosing sensitive information about participating individuals.

The problem of releasing aggregate information about a statistical database while simultaneously providing privacy to the individual participants of the database has been extensively studied in the computer science and statistical communities. There have been attempts to formalize notions of privacy in such settings and to capture the requirements of privacy in a formal model, with an ultimate goal of facilitating rigorous analyses of solutions that may be proposed as "privacy preserving". *Differential privacy* (DP) has been one of the main lines of research that has emerged out of these attempts over the last seven years. See Dwork [4] for a survey. Differential privacy formalizes the idea that privacy is provided if the privacy risk an individual faces does not change significantly if he or she participates in a statistical database.

There are numerous examples of data that have associations between entities, such as social networks, routing networks, citation graphs, biological networks, etc. Such associations between entities may be modeled as a graph, where individuals are represented by the nodes, and relationships between individuals as edges. Each node may be associated with various attributes. The risk of being identified by participating in such a database is two-fold: individuals may be identified by virtue of their attributes or they may be identified from their associations with other individuals and some background information, that they usually cannot predict or control, or they might be identified using a combination of the two. In this paper, we will only be concerned about preventing identification of the nodes using associations between individuals and some possible background information, an approach that Korolova et al. [12] call *link privacy*. Using the work of Hay et al. [9], this can be extended to include a weak form of *node privacy*. Our proposed mechanism for synthetic graph generation, which aims to approximate certain statistics of the original graph, satisfies the rigorous definition of ε-differential privacy. Private estimation of the Stochastic Kronecker Graph (SKG) model parameter is an interesting problem, especially given the surge in the popularity of SKGs for graph modeling. Our initial attempt at private SKG estimation [17] was based on the Kronecker graph estimator of Leskovec and Faloutsos [15]. However, our solution was inefficient because we did not have any practical way of bounding the *global sensitivity* of

PAIS 2012, March 30, 2012, Berlin, Germany.

the required parameters. Subsequent work by Gleich and Owen [8], which estimates the graph model parameters by using a "moment matching" method, makes it possible for us to apply the work of Hay et al. [9] and Nissim et al. [18] to efficiently compute private approximations of the "matching statistics" and, hence, to obtain private estimates of the model parameter.

To generate representative synthetic graphs, we use tools from statistical inference. Assuming that observed data is generated from an underlying, but unknown, probability distribution, we use the data to infer the distribution. A graph $G(V, E)$ is represented as a vector of random variables $\{E_1, E_2, \ldots, E_N\}$, where each of the E_i's are 0-1 random variables representing the presence or absence of an edge (assuming a specific known ordering of all potential edges between $|V|$ vertices). We assume that data is generated from a parameterized family of probability distributions. Given a graph, that is treated as a sequence of observations in such a model, our goal is to infer the parameter of the distribution and hence the distribution itself. If the estimator preserves differential privacy and is a good estimator, we can publish it and anyone interested in studying statistical properties of the original graph G can sample the distribution to yield a synthetic graph G_S which mimics the statistical properties of G. We could also sample several graphs from the distribution and compute an average of the desired statistic over several such graphs. To use such an approach, we need to impose a relevant model on the kinds of graphs we are interested in. The choice of a model is typically guided by empirical and theoretical considerations of how well the model captures key properties of real-world graphs. For our purpose, we use Leskovec et al.'s Kronecker graph model [15, 14] that effectively models salient features of real-world graphs. We compute an estimator, based on Gleich and Owen's non-private estimator [8], that is provably differentially private and that still favorably compares with the estimators proposed by Leskovec et al. [14] and Gleich and Owen [8] in terms of matching several statistics of the original graph.

Section 2 summarizes related work in privacy and anonymization. In Section 3, we provide the required background about the stochastic Kronecker graph (SKG) model and parameter estimation in this model. In Section 4, we discuss our main results: we show how we can compute an estimator of a given graph in the SKG model in a differentially private manner and also experimentally demonstrate how well the private estimator does on mimicking statistical properties of the original graph when compared to non-private methods such as those of Gleich and Owen [8] and Leskovec et al. [14]. We observe that our private estimator performs almost similalrly to Gleich and Owen's non-private estimators, for meaningful values of the privacy parameter ε.

2. RELATED WORK IN PRIVACY AND ANONYMIZATION

The problem of anonymizing databases has been receiving considerable attention over the last decade. However researchers have only recently started looking at the problem of privacy preservation in graphs. Backstrom, Dwork, and Kleinberg [1] describe a family of attacks where access to a naively anonymized graph with the identifiers of the nodes stripped can enable an adversary to learn whether edges exist or not between specified pairs of nodes. Many solutions assuming various models of attacks have been proposed: see [12, 2, 10, 23, 22] for examples. Most of that work, provides guarantees only against a specific set of adversaries who are assumed to have specific background knowledge. In reality, however, individuals and even organizations managing the database have little or no control over auxiliary information available to the adversary. ε-Differential privacy provides a guarantee that even if a participating individual removed his or her data from the database, the potential outputs of a querying mechanism (whether interactive or non-interactive) and consequences of those outputs would not become significantly more or less probable. These relative probabilities are parametrized by ε, a public parameter. We review the formal definitions introduced by Dwork et al. [6] in Section 4. While ε-differential privacy is not an absolute guarantee, it is very strong (what Dwork [4] calls an *ad omnia* guarantee), where no assumptions about the auxiliary information or computational power available to the attacker have been made.

Following the introduction of the DP framework in [3], a growing line of work has emerged identifying differentially private mechanisms for both interactive and non-interactive settings. For example, see [16, 18, 6, 4].

We attempted to use the method of using stochastic graph models to generate private "synthetic" graphs [17] but were unable to provide a useful upper bound on the *global sensitivity* of the Maximum Likelihood Estimator or an approximation of the MLE of the Stochastic Kronecker graph model used by Leskovec et al. [15]. Consequently, we were unable to use those ideas to actually model a real-world graph and release the estimator in a differentially private manner. However subsqequent work by Gleich et al. [8] estimates model parameters using a moment matching method rather than an approximation of the MLE. The algorithm matches four statistics of the observed graph to the expected values of these statistics over the probability distribution on graphs defined by the parameters. This enables us to use the work of Hay et al. [9] and the results of Nissim et al. [18] to compute differentially private approximations to these features F of the observed graph that we seek to match. Hay et al. [9] compute a differentially private approximation to the degree distribution of a graph using post-processing techniques. Nissim et al. [18] compute a differentially private approximation to the number of triangles of a graph.

Recently, Karwa et al. [11] apply the notion of *smooth sensitivity* formulated by Nissim et al. [18] to compute differentially private approximations to other graph statistics such as the number of k-triangles and k-stars. Sala et al. [20] also generate synthetic graphs that are similar to the original graph by extracting the original graph's detailed structure into degree correlation statistics, and then computing differentially private approximations of these statistics to generate a private synthetic graph. This is closest in spirit to our work. We have not yet compared the qualitiy of their private synthetic graphs to ours.

3. PARAMETRIC MODELS AND ESTIMATION

This section provides background on parametric model estimation, the Stochastic Kronecker Graph Model [15, 14] and the moment based estimation method of Gleich [8] in which our work is grounded.

A parametric statistical model, say $\mathcal{F}$, is a set of probability distributions that can be parametrized by a finite set of parameters. Parametric estimation in such a model assumes that data observed is generated from a parametrized family of probability distributions $\mathcal{F} = \{f(x;\theta) : \theta \in \Theta)\}$, where θ is an unknown parameter (or vector of parameters) that can take values in the parameter space Θ. Let $X = (X_1, X_2, \ldots X_N)$ denote N random variables representing observations $X_1 = x_1, X_2 = x_2, \ldots, X_N = x_N$, and let the joint probability density function of $(X_1, \ldots, X_N)$, given by $f(x_1, x_2, \ldots, x_N; \theta)$, depend on θ, the parameter of the distribution.

After observing this data, an estimate $\hat{\theta}$ of the unknown true parameter θ is formed. $\hat{\theta}$ is a function of the observations and hence, it is also a random variable. The problem of parameter estimation is to pick a $\hat{\theta}$ from the parameter space that best estimates the true parameter in some optimum sense. Parameter estimation is a well studied branch of statistics; see [21] for a review.

As mentioned before, the choice of a generative parametric model for graphs is typically based on empirical or theoretical considerations of how well a model captures significant descriptive properties of graphs, such as degree distribution, specific patterns observed, etc. Once such a model is defined, the task consists of estimating the parameter of the model that generated a particular instance G. G can be looked at as a sequence of observations $E_1, \ldots, E_N$ where the E_i's are 0-1 random variables representing the absence or presence of an edge i (according to a specific ordering). The estimated parameter defines a probability distribution on all graphs, one from which we assumed G was generated. One can then sample this probability distribution to generate a synthetic graph G_S and run queries on it to get an approximation to the answers that would have obtained from the original graph G. In this section, we introduce the Stochastic Kronecker Graph (SKG) model, the specific generative model we use. In Section 4, we show how to estimate the parameter in a differentially private manner that demonstrates experimental utility with respect to certain statistics.

3.1 Kronecker graph model

Modeling graphs in general, and networks in particular, is an important problem. Most work in graph modeling consists of studying patterns and properties found in real-world graphs and then finding models that help understand the emergence of these properties. Some of the key properties studied are degree distribution, diameter, hop-plot, scree plot, and node triangle participation [15, 14]. The Kronecker graph model effectively captures some of the salient patterns of real-world graphs, such as heavy tailed in-degree and out-degree distributions, heavy tails for eigenvalues and eigenvectors, small diameters, and "densification power law" observed in the Internet, the Web, citation graphs, and online social networks. Many models in the literature focus on modeling one static property of the network model while neglecting others. Moreover, the properties of many such network models have not been formally analyzed. Leskovec et al.'s Kronecker graph model has been empirically shown to match multiple properties of real networks. It also facilitates formal analysis of these properties and establishes, empirically and analytically, that Kronecker graphs mimic some important properties of real-world graphs such as those described above. The Kronecker graph results of Leskovec et al. [14, 15] have three important contributions:

1. Their graph generation model provably produces networks with many properties often found in real-world graphs, such as a power-law degree distribution and small diameter.
2. Their approximate MLE algorithm is fast and scalable, being able to handle very large networks with millions of nodes.
3. The estimated parameter generates realistic looking graphs that empirically match the statistical properties of the target real graphs.

Kronecker graphs are based on a recursive construction, with an aim of creating self-similar graphs recursively. The process starts with an *initiator graph* G_1 with N_1 nodes. By a recursive procedure, larger graphs $G_2, \ldots, G_n$ are generated in succession such that the kth graph, G_k, has $N_k = N_1^k$ nodes. This procedure is formalized by introducing the concept of Kronecker product of the adjacency matrices of two graphs [15].

DEFINITION 3.1 ([15]). *Given two matrices* $\mathbf{A}$ *and* $\mathbf{B}$ *of sizes* $n \times m$ *amd* $n' \times m'$ *respectively, their* Kronecker product *is a matrix* $\mathbf{C}$ *of dimensions* $(n \cdot n') \times (m \cdot m')$ *defined as:*

$$C = \mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{1,1}\mathbf{B} & a_{1,2}\mathbf{B} & \ldots & a_{1,m}\mathbf{B} \\ a_{2,1}\mathbf{B} & a_{2,2}\mathbf{B} & \ldots & a_{2,m}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1}\mathbf{B} & a_{n,2}\mathbf{B} & \ldots & a_{n,m}\mathbf{B} \end{pmatrix}$$

The Kronecker product of two graphs is the Kronecker product of their adjacency matrices, defined as:

DEFINITION 3.2 ([15]). *Let* G *and* H *be graphs with adjacency matrices* $A(G)$ *and* $A(H)$ *respectively. The Kronecker product* $G \otimes H$ *of the two graphs is the graph whose adjacency matrix is the Kronecker product* $A(G) \otimes A(H)$.

Informally, the Kronecker product of two graphs G and H is the "expanded" graph obtained by replacing each node in G by a copy of H. G_2 is obtained by taking the Kronecker product of G_1 with itself, G_3 by taking the Kronecker product of G_2 with G_1, and so on, such that the kth Kronecker power of G_1 gives G_k. Formally:

DEFINITION 3.3 (KRONECKER POWER [15]). *Given a Kronecker initiator adjacency matrix* Θ_1*, the kth* power of

Θ_1 *defined by*

$$\Theta_1^{[k]} = \underbrace{\Theta_1 \otimes \Theta_1 \otimes \ldots \otimes \Theta_1}_{k \text{ times}} = \Theta_1^{[k-1]} \otimes \Theta_1$$

The graph G_k *defined by* $\Theta_1^{[k]}$ *is a* Kronecker graph of order k *with respect to* Θ_1.

3.2 Stochastic Kronecker graph model

In this section we review the SKG model and in Sections 3.3 and 3.4, we review parameter estimation in this model.

Leskovec et al. [15] introduce stochasticity in the Kronecker graphs model by letting each entry of the $N_1 \times N_1$ initiator matrix Θ_1 take values in the range $[0, 1]$ instead of binary values, representing the probability of that edge being present. If the Kronecker power of Θ_1 is computed in the manner explained above, larger and larger stochastic adjacency matrices are obtained where each entry represents the probability of that particular edge appearing in the graph. $\Theta_1^{[k]}$, therefore, defines a probability distribution on all graphs with N_1^k nodes. To obtain a stochastic Kronecker graph (SKG), an edge is independently chosen with a probability specified by the corresponding entry in the matrix.

DEFINITION 3.4 (SKG). *If* Θ *is an* $N_1 \times N_1$ *probability matrix such that* $\theta_{ij} \in \Theta$ *denotes the probability that edge* (i, j) *is present,* $\theta_{ij} \in [0, 1]$. *Then the* k*th Kronecker power* $P = \Theta^{[k]}$, *is a stochastic matrix where each entry* $P_{uv} \in P$ *encodes the probability of edge* (u, v) *appearing. This stochastic matrix encodes a* stochastic Kronecker graph. *To obtain a graph* G^*, *an instance or realization of the distribution, denoted as* $R(P)$, *an edge* (u, v) *is included in* $G^* = R(P)$ *with probability* P_{uv}.

Given a stochastic matrix P, and a graph G^* realized from P in the manner specified above, each edge (i, j) in G^* is picked independently by tossing a coin with a bias specified by P_{ij}.

Notice that, G^* as defined is a directed graph, but in this paper, like Gleich et al. [8] we examine modeling of undirected graphs only. If A^* is the adjacency matrix of G^*, then it may contain loops and may not necessarily be symmetric. These loops and the assymetry are removed by defining the random graph G with an adjacency matrix A such that, $A_{ij} = 0, \forall i = j$ and symmetrizing A^* by letting $A_{ij} = A^*_{i,j}$ if $i > j$ and having $A_{ji} = A^*_{ji}$ if $i < j$.

3.3 Parameter Estimation in the SKG Model

For every graph G, $P(G)$ is the probability that a given stochastic graph model, with a given set of parameters, generates graph G. In the stochastic Kronecker graph model, probability distributions over graphs are parametrized by the initiator matrix, Θ of size $N_1 \times N_1$. An appropriate size for N_1 is decided upon using standard techniques of model selection. Analysis in [15] shows that for many real-world graphs, having $N_1 > 2$ does not accrue a significant advantage as far as matching of some statistics is concerned. In this paper, we set $N_1 = 2$, to compare our results to those obtained by Gleich et al. [8].

Given a graph G that is assumed to be generated by an SKG model, we want to estimate the true parameter—the initiator matrix Θ—that generated G by an appropriate $\hat{\Theta}$. Leskovec et al. provide an algorithm that is linear in the number of edges to estimate the parameter $\hat{\Theta}$. Let G have N nodes and assume $N = N_1^k$, where the size of the initiator matrix is $N_1 \times N_1$. Using $\Theta^{[k]} = P$, P defines a SKG on N nodes: P_{uv} is the probability that there is an edge between nodes u and v. Hence, the probability $p(G|\Theta) = p(G = R(P))$ that G is a realization of P can be computed easily. The Maximum Likelihood Estimator $\hat{\Theta}$ maximizes the likelihood of realizing G. Formally, the MLE solves:

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmax}}\, p(G|\Theta)$$

3.4 Moment based estimation of SKG's

Gleich and Owen [8] propose an alternative method to estimate SKG model parameters. They do so for reasons of computational cost of estimating the MLE of the SKG model. Leskovec et al. [14] try to approximate the MLE. Gleich and Owen use the so-called moment-based estimation of the model parameter, where the observed values of certain statistics of the graphs are equated with those of the expected value of these statistics over graphs that a paramter would define. They remark that "while moment methods can be statistically inefficient compared to maximum likelihood, statistical efficiency is of reduced importance for enormous samples and in settings where the dominant error is lack of fit."

Four statistics for matching, in the sense explained above, are considered: number of edges (E), number of triangles(Δ), number of hairpins (2-stars or wedges)(H) and the number of tripins (3-stars)(T). They consider graphs with a 2×2 initiator matrix of the form

$$\Theta = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$$

with $a, b, c, \in [0, 1]$ and $a \geq c$. The Kronecker structure of P makes it possible to compute closed formulae for these statistics from Θ. Given Θ of the form above, the expected count for these statistics can be calculated explicitly. Specifically, given $P = [\Theta]^k$, closed formulae can be derived for H, Δ and T in terms of a, b, c, as follows:

$$\begin{aligned}
\mathbb{E}(E) =& \frac{1}{2}\left((a+2b+c)^k - (a+c)^k\right) \\
\mathbb{E}(H) =& \frac{1}{2}\Big(\left((a+b)^2 + (b+c)^2\right)^k - 2\left(a(a+b) + c(c+b)\right)^k \\
& - (a^2+2b^2+c^2)^k + 2(a^2+c^2)^k\Big) \\
\mathbb{E}(\Delta) =& \frac{1}{6}\Big(((a^3+3b^2(a+c)+c^3)^k - 3a(a^2+b^2)+ \\
& c(b^2+c^2))^k + 2(a^3+c^3)^k\Big) \\
\mathbb{E}(T) =& \frac{1}{6}\Big(\left((a+b)^3 + (b+c)^3\right)^k - 3\left(a(a+b)^2 + c(b+c)^2\right)^k \\
& -3(a^3+c^3+b(a^2+c^2)+b^2(a+c)+2b^3)^k + 2(a^3+ \\
& 2b^3+c^3)^k + 5(a^3+c^3+b^2(a+c))^k \\
& +4(a^3+c^3+b(a^2+c^2))^k - 6(a^3+c^3)^k\,)
\end{aligned} \tag{1}$$

The problem then is to find an initiator matrix whose expected counts match the counts of the features $F(G)$ of the observed graph as closely as possible.

Given G, one way to choose $\hat{\Theta}$ (or equivalently, $\hat{a}, \hat{b}$, and $\hat{c}$) is to solve

$$\min_{a,b,c} \sum_{F} \frac{(F - \mathbb{E}_{a,b,c}(F))^2}{\mathbb{E}_{a,b,c}(F)}$$

where the sum is over three of four of the features $F \in \{E, \Delta, H, T\}$ and the minimization is taken over $0 \leq c \leq a \leq 1$ and $0 \leq b \leq 1$. A more general minimization method solves:

$$\min_{a,b,c} \sum_{F} \frac{\text{Dist}(F, \mathbb{E}_{a,b,c}(F))}{\text{Norm}(F, \mathbb{E}_{a,b,c}(F))}, \tag{2}$$

where Dist is either of the two distance functions:

$$\text{Dist}_{\text{Sq}}(x, y) = (x - y)^2 \text{ or } \text{Dist}_{\text{abs}}(x, y) = |x - y|$$

and Norm is one of the normalizations:

$\text{Norm}_F = (F, \mathbb{E}) = F; \text{Norm}_{F^2}(F, \mathbb{E}) = F^2$; $\text{Norm}_{\mathbb{E}}(F, \mathbb{E}) = \mathbb{E}; \text{Norm}_{\mathbb{E}^2}(F; \mathbb{E}) = \mathbb{E}^2$.

Gleich and Owen [8] find that robust results arise from the combination of Dist_{Sq} and Norm_{F^2}. The next section uses these results.

4. A DIFFERENTIALLY PRIVATE GRAPH ESTIMATOR

We present our main result in this section. We use the results of Gleich and Owen [8] to provide a differentially private estimator of a given graph. Based on experimental results, in Section 4.2 we argue that a modification that makes the estimator differentially private does not destroy the desirable properties of the graph model estimator for both some real-world and synthetic networks.

4.1 Differential Privacy

After a private estimator is computed, we may publish it and sample graphs from this distribution to compute an approximation of relevant statistics. Under the assumption that the model captures the essential properties of the graph, our estimator will define a probability distribution from which we can sample graphs that are "similar" to the original graph G. We emphasize here that we rely upon the results of [14] to justify using the SKG model to maintain "similarity" of synthetic graphs to the original graphs. Our private estimator suffers from the same limitation that the SKG does in capturing properties of a real-world network but also demonstrates almost the same accuracy. In this section, we present our main result showing how to compute an estimator for the SKG model that is also differentially private. We first formalize the idea that the output of the estimator should not change significantly if a link between two individuals is included or excluded from the observations.

Definition 4.1 (Edge neighborhood [9, 17]). *Given a graph $G(V, E)$, the (edge)* neighborhood of a graph *is the set*

$$\Gamma(G) = \{G'(V, E') \; s.t \; |E \oplus E'| = 1\}$$

Applying the standard definition of differential privacy to graphs instead of databases and using the above definition of neighborhood yields the following:

Definition 4.2 (Edge differential privacy [18]). *A parameter estimation algorithm that takes as input a graph G, and outputs $\tilde{\Theta}(G)$, preserves (ε, δ)*-differential edge privacy *if for all closed subsets S of the output parameter space, and all pairs of neighboring graphs G and G', and for all $\delta \in [0, 1]$,*

$$\Pr[\tilde{\Theta}(G) \in S] \leq \exp(\varepsilon) \cdot \Pr[\tilde{\Theta}(G') \in S] + \delta$$

The original notion of ε-differential privacy [6] is a special case of the (ε, δ)-differential privacy in which $\delta = 0$.

Hay et al. [9] also define *node differential privacy*, by analogously defining the notion of *node neighborhood* of a graph. Two graphs are node neighbors if they differ by at most one node and all the incident edges. This notion of privacy is highly restrictive when trying to compute accurate approximations of graph statistics because of potentially high degree nodes and the loss of information that would accompany their deletion. To provide some degree of privacy to nodes, Hay et al. [9] introduce the notion of *k-edge differential privacy*. In k-edge differential privacy, graphs G and G' are *k-edge neighbors* if $|V \oplus V'| + |E \oplus E'| \leq k$. They also make the observation that any algorithm that provides ε-edge privacy with respect to (1-)edge neighbors, will provide $k\varepsilon$-edge privacy with respect to k-edge neighbors using a well-known composition theorem (stated here as Theorem 4.9). In this paper, we only examine 1-edge differential privacy.

According to Definition 4.2, for a graph estimator that preserves differential privacy, outputs of the estimating algorithm do not become significantly more or less likely if an edge is included or excluded from the database. If the inclusion or exclusion of a single link between individuals cannot change the output distribution appreciably, even an adversary who may have additional background information will not, by interacting with the algorithm, learn significantly more about an individual than could be learned about this individual otherwise.

Dwork et al. [6] and Nissim et al. [18] define the notions of *local sensitivity* and *global sensitivity*:

Definition 4.3 (Local Sensitivity [18]). *The* local sensitivity *of $f : D \to \mathbb{R}$, that maps a Domain D to reals, at $G \in D$ is*

$$\text{LS}_f(G) := \max_{G' \; s.t. \; G' \in \Gamma(G)} \|f(G) - f(G')\|_1$$

As an example, when computing the local sensitivity of the number of triangles in a graph G having N nodes, the domain D is the space of all graphs on N nodes.

Definition 4.4 (Global Sensitivity [6]). *The* global sensitivity *of a function of a graph G, $f : D \to \mathbb{R}^\ell$ is*

$$\text{GS}_f := \max_{G \in D} \; \text{LS}_f(G)$$

Using these notions we compute a differentially private estimator based on matching the expected count to the observed counts of the statistics—we supply differentially private approximations of the statistics E, H, Δ and T to Equation 2. We do this by computing differentially private approximation to the degree sequence vector of G and the number of triangles in G.

Let d be the vector of degrees of G, such that d_i is the degree of node i of graph G. Let d be sorted to yield d_S such that $d_S(i)$ is the i-th smallest degree. Hay et al. [9] propose a method of computing a differentially private approximation $\tilde{d}$ of the sorted degree vector d_S by adding a vector of appropriate Laplacian noise to d_S and then using post-processing techniques that they experimetally show to be a highly accurate approximation of d_S. Let $\langle \text{Lap}(\sigma) \rangle^N$ denote a N length vector of independent random samples from a Laplace distribution with mean zero and scale σ. We know that the global sensitivity of d_S, GS_d is equal to 2. The Laplacian noise adding method os as follows:

THEOREM 4.5 ([6]). *Let $\hat{Q}$ denote the randomized algorithm that takes as input graph G, a query Q of length ℓ, and some $\varepsilon > 0$, and outputs*

$$\hat{Q}(G) = Q(G) + \langle \text{Lap}(\text{GS}_Q / \varepsilon) \rangle^{\ell}.$$

Then, Algorithm $\hat{Q}$ satisfies $(\varepsilon, 0)$-differential privacy.

Hay et al. [9] use Theorem 4.5 to compute a "noisy" degree sequence $\hat{d}$ as an approximation of d_S:

$$\hat{d} = d_S + \langle \text{Lap}(2/\varepsilon) \rangle^N.$$

Therefore, $\hat{d}$ is then an $(\varepsilon, 0)$-differentially private approximation of d_S. Hay et al. [9] use post-processing techniques that seek to "remove some of the extra noise" in $\hat{d}$, to compute a $\tilde{d}$ that is experimentally and theoretically shown to provide higher accuracy. Using $\tilde{d}$, we compute $(\varepsilon, 0)$-differentially private approximations of E, H, and T in the following manner:

$\tilde{E} = \frac{1}{2}\sum_i \tilde{d}_i$; $\tilde{H} = \frac{1}{2}\sum_i \tilde{d}_i(\tilde{d}_i - 1)$ and $\tilde{T} = \frac{1}{6}\sum_i \tilde{d}_i(\tilde{d}_i - 1)(\tilde{d}_i - 2)$. Hence, we have:

FACT 4.6. *Computing $\tilde{E}, \tilde{H}$ and $\tilde{T}$ using $\tilde{d}$ is $(\varepsilon, 0)$-differentially private.*

This is straightforward, as computing $\tilde{d}$ is $(\varepsilon, 0)$-differentially private. Since the number Δ of triangles is not a simple function of the degree distribution, we instead use the tehcniques of Nissim et al. [18] to compute an (ε, δ)-differentially private approximation of Δ. To reduce the amount of noise that needs to be added to compute an approximation to Δ, Nissim et al. [18] use an upper bound on the local sensitivity of $\Delta(G)$ by computing the *β-smooth sensitivity* of $\Delta(G)$.

Let $\text{dist}(G, G')$ be the symmetric difference between the edge sets of graphs G and G'. Hence, if G and G' are neighbors by Definition 4.1, $\text{dist}(G, G') = 1$.

DEFINITION 4.7 (β-SMOOTH SENSITIVITY [18]). *For $\beta > 0$, the β-smooth sensitivity of f at G, is*

$$\text{SS}_{\beta,f}(G) = \max_{G'} \left(\text{LS}_f(G) \cdot e^{-\beta\, \text{dist}(G;G)} \right)$$

The smooth sensitivity can be used to compute a differentially private approximation to a function f:

THEOREM 4.8 ([18]). *Let $f : D^n \to R$ be any real-valued query function from an input $x \in D^n$ for some domain D, and let $\text{SS}_{\beta,f} : D^n \to R$ be the β-smooth sensitivity of f for some $\beta > 0$. Then, if $\beta < \frac{\varepsilon}{2\ln(2/\delta)}$ and $\delta \in (0, 1)$, the algorithm that outputs $\tilde{f} = f(D) + 2\frac{\text{SS}_{\beta,f}(D)}{\varepsilon}.\eta$, where $\eta \sim \text{Lap}(1)$, is (ε, δ)-differentially private.*

Algorithm 1 illustrates the process we adopt. Our results use the above theorem and a composition theorem:

THEOREM 4.9 (COMPOSITION THEOREM [5]). *Let $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_\ell$, be ℓ number of (ε, δ)-differentially private mechanisms computed using graph G. Then any mechanism $\mathcal{M}$ that is a composition of $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_\ell$, is $(\ell\varepsilon, \ell\delta)$-differentially private.*

Using these results we compute an (ε, δ)-differentially private approximation of Δ by outputting:

$$\tilde{\Delta} = \Delta + 2\frac{\text{SS}_{\beta,\Delta}\, 2}{\varepsilon}.\, \text{Lap}(1),$$

as an (ε, δ)-differentially private approximation to the number of triangles in G. Using Theorems 4.9, 4.8, and Fact 4.6, we have

THEOREM 4.10. *The computation of $\tilde{F} = \{\tilde{E}, \tilde{H}, \tilde{T}, \tilde{\Delta}\}$ is $(2\varepsilon, \delta)$ differentially private.*

Using these private statistics $\tilde{F}$, in the moment-matching algorithm of Gleich and Owen (Equation 2), we obtains a differentially private estimator. Algorithm 1 illustrates the process. Hence, we have:

COROLLARY 4.11. *$\tilde{\Theta}$ computed by Algorithm 1 is (ε, δ)-differentially private.*

4.2 Experimental results

In this section, we discuss application of Algorithm 1 to three real-world networks and two synthetic Kronecker graphs. CA-GrQC and CA-HepTh are co-authorship networks from arXiv [14]. The nodes of the network represent authors, and there is an edge between two nodes when the authors jointly wrote a paper. AS20 is a real-world technological infrastructure network [14]. Each node represents a router on the internet and edges represent a physical or virtual connection between the routers. All these graphs are naturally undirected and all edges are unweighted. We downloaded these networks from Snap [13] and used the provided

Algorithm 1 Differentially-private estimation of $\tilde{\Theta}$

Input: Graph G, privacy parameters (ε, δ)

1. Compute the degree vector d of G.
2. Using Hay et al. [9] compute a $\varepsilon/2$-differentially private approximation of d, $\tilde{d}$
3. Compute $\tilde{E}, \tilde{H}, \tilde{T}$ from $\tilde{d}$.
4. Compute the smooth sensitivity $\mathrm{SS}_{\beta,\Delta}$ of Δ
5. Use $\mathrm{SS}(G)$ to compute an $(\varepsilon/2, \delta)$ private approximation of Δ, $\tilde{\Delta}$.
6. Use the Kronecker Moment Estimation of [8] with $\{\tilde{E}, \tilde{H}, \tilde{T}, \tilde{\Delta}\}$ as inputs to Equation 2 to compute $\tilde{\Theta}$.

Output: $\tilde{\Theta}$

Network	**KronFit** (a, b, c)	**KronMom** (a, b, c)	**Private** (a, b, c)
CA-GrQC	0.999 0.245 0.691	1.000 0.4674 0.2790	1.000 0.4618 0.2930
CA-HepTh	0.999 0.271 0.587	1.000 0.4012 0.3789	1.000 0.4048 0.3720
AS20	0.987 0.571 0.049	1.000 0.6300 0.000	1.000 0.6286 0.000
Synthetic $\Theta =$ [.99.45; .45.25]	0.9523 0.4743 0.2493	0.9894 0.5396 0.2388	0.9924 0.5343 0.2466

Table 1: Comparison of parameter estimation for $\varepsilon = 0.2, \delta = 0.01$

library for our experiments. We also used the code provided by Gleich [7] to compute both the private and non-private moment-based estimators of the networks. Table 1 compares the results of Algorithm 1 (column titled "Private") to those of Gleich et al. [8] ("KroMom") and Leskovec et al. [14] ("KronFit"). Our results are based on Gleich et al.'s results, so it is not surprising that our results are close to theirs—we observe that the private parameters we compute are very similar. To provide a reasonable comparison, for each of the graphs, we use the same Dist and Norm functions in the parameter estimation of Equation 2 as Gleich and Owen.

For the synthetic Kronecker graph we start with an initiator matrix

$$\Theta = \begin{pmatrix} 0.99 & 0.45 \\ 0.45 & 0.25 \end{pmatrix}$$

and $k = 14$ to obtain a synthetic graph on 2^{14} nodes. Then we try to recover the parameters of this synthetic graph by running all three algorithms on it. From Table 1, we see that all three algorithms do a satisfactory job in recovering the parameter when the modeling assumption is true, that is when the graph indeed is a stochastic Kronecker graph.

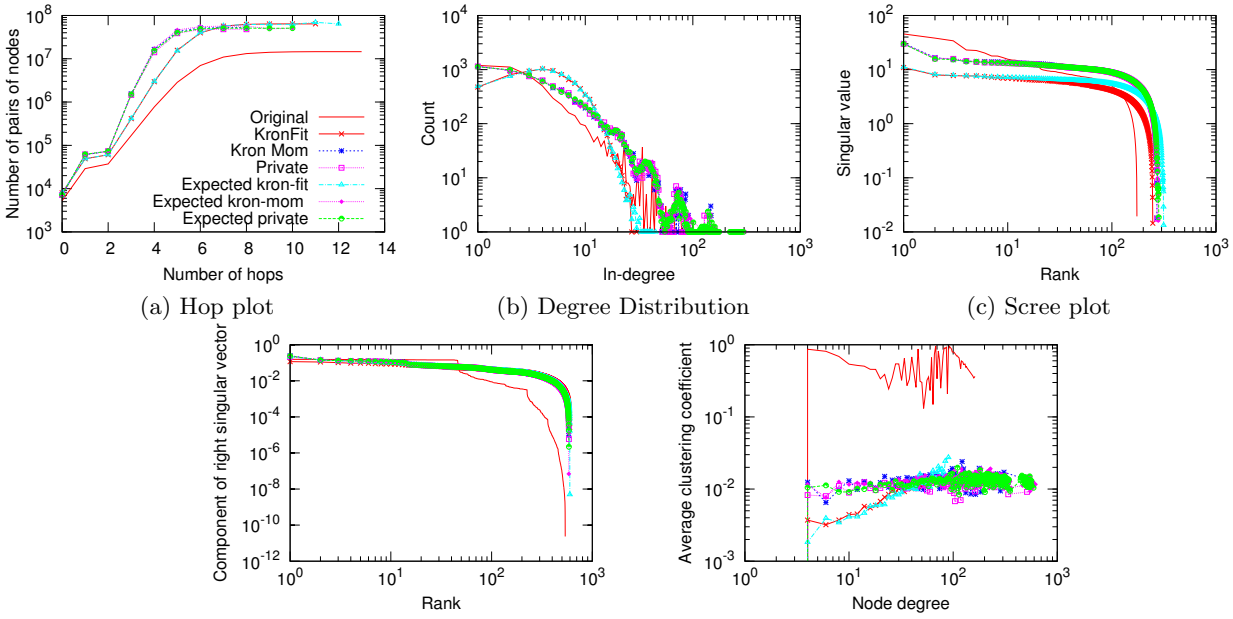
To further understand how well the private estimator captures various properties of the graph, we carry out further experiments. All experiments are conducted for $(0.2, 0.1)$-differential privacy. Using the parameter estimates of a graph, we generate 100 synthetic graphs from the estimated parameters for all three methods, and compute various expected statistics over these 100 graphs. These statistics have been computed in [14] for these graphs, so we compare the performance of our private estimator on these statistics to Leskovec et al.'s results. We summarize these statistics briefly:

1. The *degree distribution* plots the distribution of the degrees of the nodes.
2. The *Hop-plot* plots the number of reachable pairs of noded within h hops, as a function of the number of hops h.
3. The *Scree plot* plots the eigenvalues (or singular values) of the graph adjacency matrix, versus their rank, using the logarithmic scale.
4. The *Network values* plots the distribution of eigenvector components (indicators of "network value") associated with the largest eigenvalue of the graph adjacency matrix.
5. The *average clustering coefficient* plotted as a function of the node degree. The clustering coefficient is a measure of the extent to which nodes in a graph tend to cluster together.

For each of these graphs we plot these statistics. "*Original*" refers to the original graph, "*KronFit*" refers to a single synthetic Kronecker graph generated from the parameter $\hat{\Theta}$ which is computed using the KronFit algorithm of Leskovec et al. [15]. "*KronMom*" refers to a single synthetic Kronecker graph generated from the parameter $\hat{\Theta}$ that is computed using the "*KronMom*", moment-matching algorithm of Gleich and Owen [8]. "*Private*" refers to a single Kronecker graph generated from $\tilde{\Theta}$ computed in Algorithm 1. The prefix "*Expected*" refers to the expected value of the statistics being computed over 100 synthetic realizations of the appropriate Kronecker graphs.

From Fig 1, we notice that the observed statistics for a single realization are very close to the expected values, hence one realization appears to give us a representative sample, at least for these four graphs.

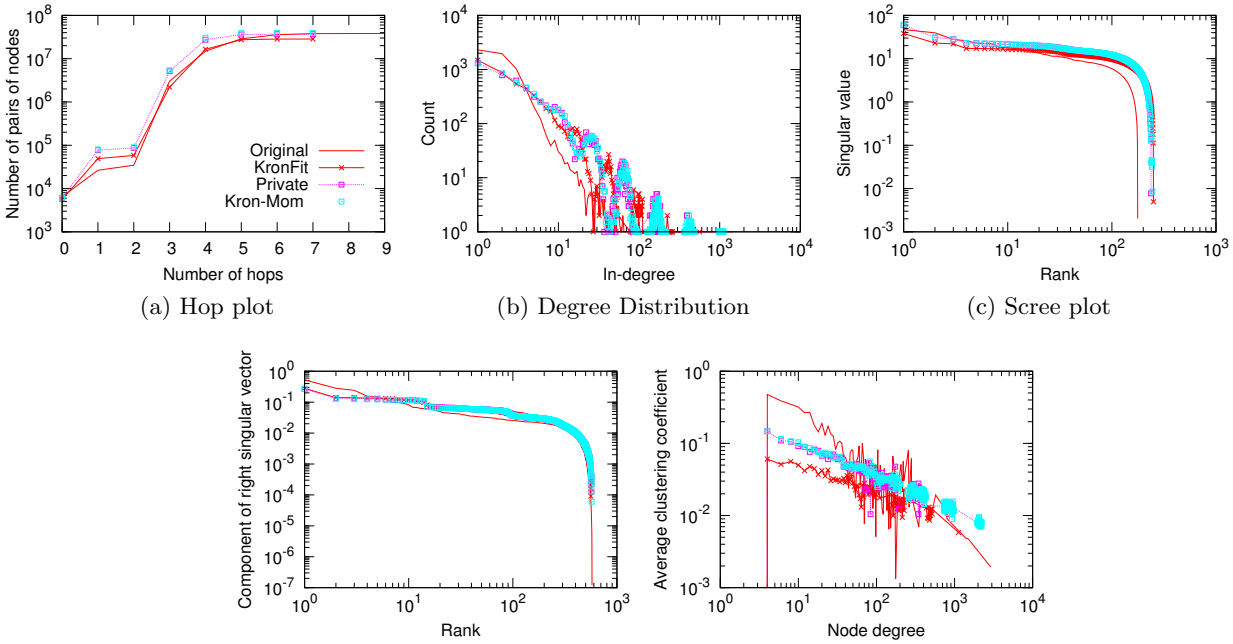
To reduce clutter, for CA-HepTh (Figure 3), AS20 (Figure 2), and the synthetic Kronecker graph (Figure 4), we only show single realizations. We observe that in all four cases, the statistics are well-approximated and very close to the "predictions" made by both the "KronFit" and the "KronMom" estimators. In the case of the synthetic Kronecker graph we also observe a good matching of the average clustering coefficient which is usually not the case for real-world networks. This has to do with modeling assumptions. We see that the SKG models the clustering coefficient well for AS20 but not for CA-GrQC and CA-HepTh. The private estimators are also observed to perform comparably.



(a) Hop plot (b) Degree Distribution (c) Scree plot

(d) Network value (e) Clustering

Figure 1: Overlayed patterns of real network for CA-GrQC ($N = 5,242, E = 28,980$) and the estimated synthetic Kronecker graph using the three different estimators.



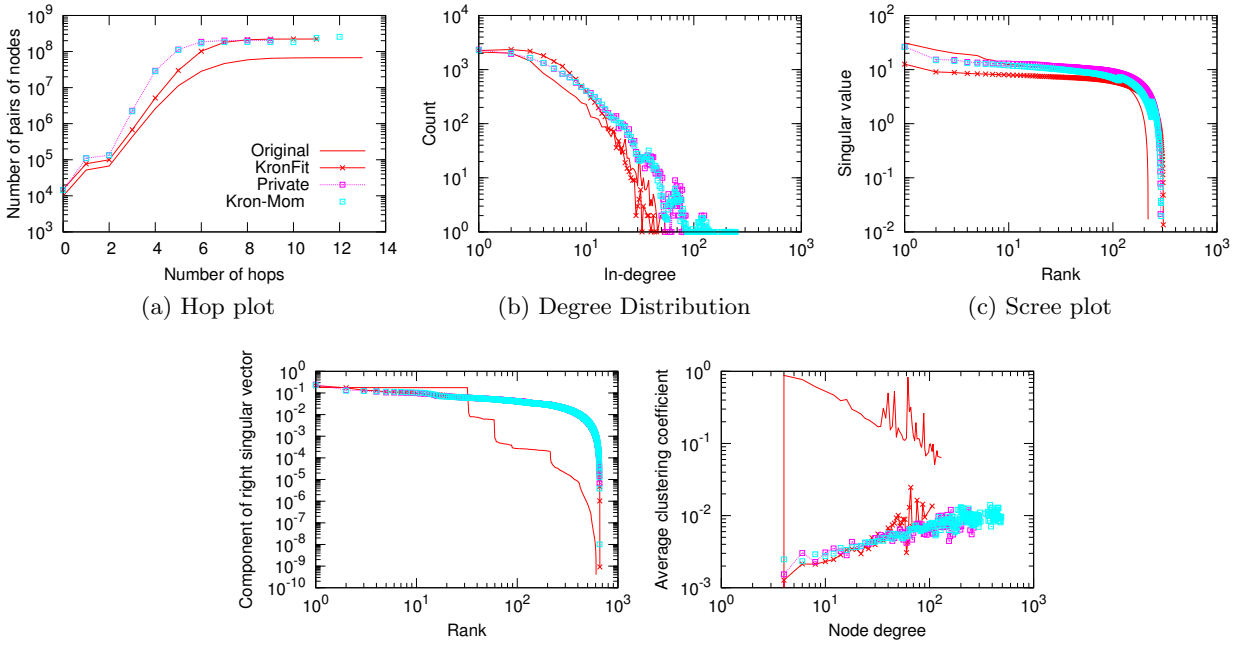
(a) Hop plot (b) Degree Distribution (c) Scree plot

(d) Network (e) Clustering

Figure 2: Overlayed patterns of real network for AS20 ($N = 6,474, \ E = 26,467$) and the estimated synthetic Kronecker graphs using the three different estimators.

5. CONCLUSIONS AND FUTURE DIRECTIONS

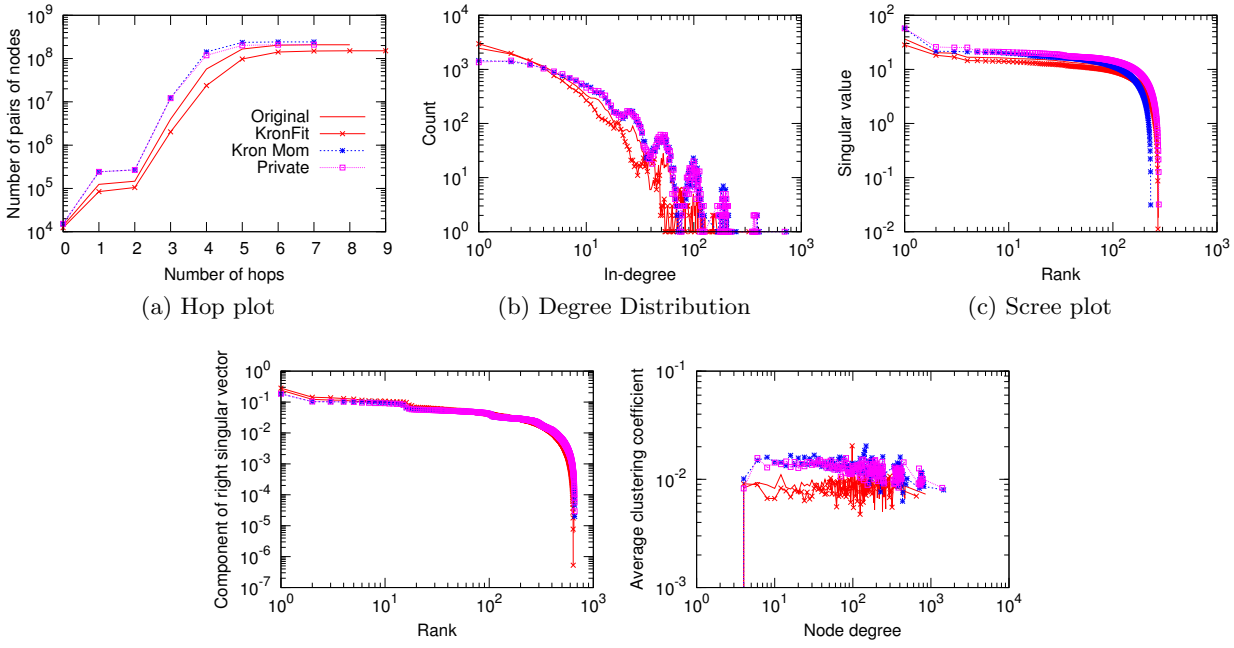
We applied the rigorous differential privacy framework to problems of generating synthetic graphs that can be made publicly available for research purposes while providing privacy to the individual participants. We built upon the work



(a) Hop plot (b) Degree Distribution (c) Scree plot

(d) Network (e) Clustering

Figure 3: Overlayed patterns of real network for CA-HepTh ($N = 9,877,\ E = 51,971$) and the estimated synthetic Kronecker graph using the three different estimators.



(a) Hop plot (b) Degree Distribution (c) Scree plot

(d) Network (e) Clustering

Figure 4: Overlayed patterns of a synthetic source Kronecker network and the estimated sythetic Kronecker graph using the three different estimators.

of Leskovec et al. [14, 15] and Gleich and Owen [8] in the generative Kronecker graph model to demonstrate that synthetic graphs that are statistically similar to the original sensitive graphs can be generated in a manner that is differentially private. While we used a specific model and a specific estimator, our work can be broadly placed in the

framework of private parametric estimation for graph models.

There are several future directions for future work. A comparison of our results to those of Sala et al. [20] seems most relevant. We plan on undertaking a study that compares the estimated statistics of the synthetic graphs derived by our method to those computed by Sala et al. An empirical study of the smooth sensitivity of the number of triangles in the SKG is another direction we are currently pursuing. Nissim et al. [18] propose an upper bound on the smooth sensitivity of the number of triangles in the $G(n, p)$ Erdos-Renyi model. It would be interesting to examine the smooth sensitivity of Δ as a function of the size of the graph G. Preliminary experiments indicate that in the SKG model, SS_Δ might grow slowly. Yet another direction that presents itself is examine private estimation in other graph models such as the *Exponential Random Graph Model* (ERGM) [19], especially since the results of Karwa et al. [11] provide accurate differentially private approximations to statistics used in ERGM estimation.

6. ACKNOWLEDGEMENTS

This work was supported by NSF award CCF-1018445. We would like to thank Geetha Jagannathan and Aleksander Nikolov for useful suggestions.

7. 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*, pages 181–190, 2007.

[2] G. Cormode, D. Srivastava, T. Yu, and Q. Zhang. Anonymizing bipartite graph data using safe groupings. In *VLDB '08: Proceedings of the 34th International Conference on Very Large Data Bases*, pages 833–844, 2008.

[3] I. Dinur and K. Nissim. Revealing information while preserving privacy. In *PODS '03: Proceedings of the 22nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 202–210, 2003.

[4] C. Dwork. Differential privacy. In *ICALP '06: Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (2)*, pages 1–12, 2006.

[5] C. Dwork and J. Lei. Differential privacy and robust statistics. In *STOC '09: Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, pages 371–380, 2009.

[6] C. Dwork, F. Mcsherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC '06: In Proceedings of the 3rd Theory of Cryptography Conference*, pages 265–284, 2006.

[7] D. Gleich. Kronecker moment based estimation code. https://dgleich.com/gitweb/?p=kgmoments;a=summary, 2011.

[8] D. F. Gleich and A. B. Owen. Moment based estimation of stochastic Kronecker graph parameters. *Internet Mathematics*, To appear.

[9] M. Hay, C. Li, G. Miklau, and D. Jensen. Accurate estimation of the degree distribution of private networks. In *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*, ICDM '09, pages 169–178, Washington, DC, USA, 2009. IEEE Computer Society.

[10] M. Hay, G. Miklau, D. Jensen, D. Towsley, and P. Weis. Resisting structural re-identification in anonymized social networks. In *VLDB '08: Proceedings of the 34th International Conference on Very Large Databases*, pages 102–114, 2008.

[11] V. Karwa, S. Raskhodnikova, A. Smith, and G. Yaroslavtsev. Private analysis of graph structure. *PVLDB*, 4(11):1146–1157, 2011.

[12] A. Korolova, R. Motwani, S. U. Nabar, and Y. Xu. Link privacy in social networks. In *CIKM '08: Proceeding of the 17th ACM Conference on Information and Knowledge Management*, pages 289–298, 2008.

[13] J. Leskovec. Snap: Stanford network analysis platform, 2010.

[14] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Gharamani. Kronecker graphs: an approach to modeling networks. arXiv:0812.4905v1, 2008.

[15] J. Leskovec and C. Faloutsos. Scalable modeling of real graphs using kronecker multiplication. In *ICML '07: Proceedings of the 24th International Conference on Machine Learning*, pages 497–504, 2007.

[16] F. McSherry and K. Talwar. Mechanism design via differential privacy. In *FOCS '07: Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, pages 94–103, 2007.

[17] D. J. Mir and R. N. Wright. A differentially private graph estimator. In *ICDM Workshops*, pages 122–129, 2009.

[18] K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. In *STOC '07: Proceedings of the 33rd Annual ACM symposium on Theory of computing*, pages 75–84, 2007.

[19] G. Robins, P. Pattison, Y. Kalish, and D. Lusher. An introduction to exponential random graph (p) models for social networks. *Social Networks*, 29(2):173–191, 2007.

[20] A. Sala, X. Zhao, C. Wilson, H. Zheng, and B. Y. Zhao. Sharing graphs using differentially private graph models. In *IMC*, Berlin, Germany, November 2011.

[21] L. Wasserman. *All of Statistics : A Concise Course in Statistical Inference (Springer Texts in Statistics)*. Springer, September 2004.

[22] E. Zheleva and L. Getoor. Preserving the privacy of sensitive relationships in graph data. In *PinKDD '07: Proceedings of the First International Workshop on Privacy, Security, and Trust in KDD*, pages 153–171, 2007.

[23] B. Zhou and J. Pei. Preserving privacy in social networks against neighborhood attacks. In *ICDE '08: Proceedings of the 24th IEEE International Conference on Data Engineering*, pages 506–515, 2008.