TRANSACTIONS ON DATA PRIVACY 6 (2013) 127–145

Preserving Differential Privacy in Degree-Correlation based Graph Generation

Yue Wang*, **Xintao Wu***

*Software and Information Systems Department, University of North Carolina at Charlotte, Charlotte, NC 28223, USA. E-mail: {ywang91,xwu}@uncc.edu

Abstract. Enabling accurate analysis of social network data while preserving differential privacy has been challenging since graph features such as cluster coefficient often have high sensitivity, which is different from traditional aggregate functions (e.g., count and sum) on tabular data. In this paper, we study the problem of enforcing edge differential privacy in graph generation. The idea is to enforce differential privacy on graph model parameters learned from the original network and then generate the graphs for releasing using the graph model with the private parameters. In particular, we develop a differential privacy preserving graph generator based on the dK-graph generation model. We first derive from the original graph various parameters (i.e., degree correlations) used in the dK-graph model, then enforce edge differential privacy on the learned parameters, and finally use the dK-graph model with the perturbed parameters to generate graphs. For the 2K-graph model, we enforce the edge differential privacy by calibrating noise based on the smooth sensitivity, rather than the global sensitivity. By doing this, we achieve the strict differential privacy guarantee with smaller magnitude noise. We conduct experiments on four real networks and compare the performance of our private dK-graph models with the stochastic Kronecker graph generation model in terms of utility and privacy tradeoff. Empirical evaluations show the developed private dK-graph generation models significantly outperform the approach based on the stochastic Kronecker generation model.

Keywords. Differential Privacy, Graph Generation, dK-graph, Kronecker Graph

1 Introduction

Omnipresent graph databases of various networks, especially social networks, have provided researchers with unprecedented opportunities to analyze complex social phenomena. While society would like to encourage such scientific endeavors, privacy disclosure, or the risk of being identified, has attracted more and more attention by users of such networks. To help guiding public policy to protect individuals' privacy as well as promoting scientific analysis of social networks, we are faced with the problem of providing researchers with a fairly precise picture of the quantities or trends of the networks without disclosing sensitive information about participants of the network.

Graph topologies play an irreplaceable role in the network analysis. Previous research in security and privacy has shown potential risks for individual identification with the real graph topologies or the anonymized graph topologies of social networks [1, 2, 3]. Various

anonymization approaches [4, 5, 6, 7] have been developed to protect privacy. However, those approaches adopt the idea of pre-processing the raw graph such that each individual or its sensitive links are hidden within a group of other individuals. They often assume that adversaries have a particular type of structural background knowledge (e.g., vertex degrees, neighborhoods, embedded subgraphs, graph metrics) in their attacks . For example, Liu and Terzi [4] considered vertex degrees as background knowledge of the adversaries to breach the privacy of target individuals, the authors of [5, 2] used neighborhood structural information of some target individuals, the authors of [1, 6] proposed the use of embedded subgraphs, and Ying and Wu [7] exploited the topological similarity/distance to breach the link privacy. Hence there is no guarantee to achieve strict privacy protection since they could not completely prevent adversaries from exploiting various auxiliary information to breach privacy.

There have been attempts [8, 9, 10, 11] to formalize notions of differential privacy in releasing aggregate information about a statistical database and the mechanism to providing privacy protection to participants of the databases. Differential privacy [8, 9] is a paradigm of post-processing the output of queries such that the inclusion or exclusion of a single individual from the data set make no statistical difference to the results found. Differential privacy provides formal privacy guarantees that do not depend on an adversary's background knowledge (including access to other databases) or computational power. One classic method to achieve differential privacy is to directly add calibrated laplace noise on the output of the computation. The calibrating process includes the calculation of the global sensitivity of the computation that bounds the possible change in the computation output over any two neighboring databases, and adds a random noise generated from a Laplace distribution with the scale parameter determined by the global sensitivity and the specified privacy threshold. This approach works well for traditional aggregate functions (often with low sensitivity values) over tabular data.

In the context of privacy for graphs, the authors in [12] introduced 1) edge differential privacy where two neighboring graphs differ at most one edge, 2) k-edge-differential privacy where two neighboring graphs can differ at most k edges, and 3) node differential privacy where two neighboring graphs can differ up to all edges connected to one single node. Node differential privacy assures more privacy concerns as a node differentially private algorithm behaves almost as if an individual did not appear in the released graph at all. While node-differential privacy is a desirable objective, it may be infeasible to design algorithms that both achieve node privacy guarantee and enable accurate graph analysis. For example, it was shown in [12] that graph analysis is highly inaccurate under node-differential privacy or k-edge differential privacy (when k is large) due to large calibrated noise in order to achieve privacy guarantee. Throughout this paper, we focus on edge differential privacy. Our motivation is to protect sensitive relationships between individuals in sharing social graph topology, where providing edge privacy would address a number of practical privacy attacks [1].

There have been attempts to enforce differential privacy on graph data, e.g., computing graph properties such as degree distributions [12], clustering coefficient [13, 14], and eigenvalues/eigenvectors [15] in social network analysis. However, it is very challenging to directly enforce differential privacy in computing graph properties (e.g., clustering coefficient) due to their high sensitivity. Recently, attempts [16, 17] have been made in enforcing edge differential privacy in graph generation. The idea is to enforce edge differential privacy on graph model parameters learned from the original network and then generate the graphs for releasing using the graph model with the private parameters. The released graphs then can be used for various analysis. The authors in [16] tried to generate differen-

tially private graph topology with the stochastic Kronecker graph generation model [18]. However, the stochastic Kronecker graph generation model often cannot accurately capture graph properties of real social networks due to its simplicity in the generation process. The authors in [17] developed a private dK-graph model to enforce edge differential privacy. The dK-graph model [19], which constructs graphs to satisfy a family of properties based on various types of node degree correlations, has been shown an effective graph generation model. However, the private 2K-graph model proposed in [17] was based on the local sensitivity of degree correlations due to the large global sensitivity. As shown in [11], the noise magnitude based on the local sensitivity reveals information about the data. As a result, the model in [17] could not achieve rigorous differential privacy protection. In this paper, we present a private 2K-graph generation model that achieves rigorous edge differential privacy. Our idea is to enforce the edge differential privacy by calibrating noise based on the smooth sensitivity [11]. The smooth sensitivity is to use a smooth upper bound on the local sensitivity when deciding the noise magnitude. By doing this, we achieve the strict differential privacy guarantee with smaller magnitude noise. We conduct experiments on four real networks and compare the performance of our private dK-graph models with the stochastic Kronecker graph generation model in terms of utility and privacy tradeoff. Empirical evaluations show the effectiveness of our proposed private dK-graph models.

2 Background

2.1 Differential Privacy

We revisit the formal definition and the mechanism of differential privacy. In prior work on differential privacy, a database is treated as a collection of *rows*, with each row corresponding to the data of a different individual. Here we focus on how to compute graph statistics from private network topology described as its adjacency matrix. We aim to ensure that the inclusion or exclusion of a link between two individuals from the graph make no statistical difference to the results found.

Definition 1. (Differential Privacy[9]) A graph analyzing algorithm Ψ that takes as input a graph G, and outputs $\Psi(G)$, preserves (ε, δ)-differential edge privacy if for all closed subsets S of the output space, and all pairs of neighboring graphs G and G' from $\Gamma(G)$,

$$Pr[\Psi(G) \in S] \leq e^{\varepsilon} \cdot Pr[\Psi(G') \in S] + \delta, \tag{1}$$

where

$$\Gamma(G) = \{G'(V, E') | \exists!(u, v) \in G \; but \; (u, v) \notin G'\}. \tag{2}$$

A differentially private algorithm provides an assurance that the probability of a particular output is almost the same whether or not any individual edge is included. The privacy parameter ϵ, δ controls the amount by which the distributions induced by two neighboring graphs may differ (smaller values enforce a stronger privacy guarantee).

A general method for computing an approximation to any function f while preserving ϵ-differential privacy is given in [8]. The mechanism for achieving differential privacy computes the sum of the true answer and random noise generated from a Laplace distribution. The magnitude of the noise distribution is determined by the sensitivity of the computation and the privacy parameter specified by the data owner. The sensitivity of a computation bounds the possible change in the computation output over any two neighboring graphs (differing at most one link).

Definition 2. (Global Sensitivity[8]) The global sensitivity of a function $f : D \rightarrow \mathbf{R}^d$ ($G \in D$),in the analysis of a graph G, is

$$GS_f(G) := \max_{G,G' s.t. G' \in \Gamma(G)} ||f(G) - f(G')||_1 \tag{3}$$

Theorem 1. (The Mechanism of Adding Laplace noise[8]) An algorithm A takes as input a graph G, and some $\varepsilon > 0$, a query Q with computing function $f : D^n \rightarrow \boldsymbol{R}^d$, and outputs

$$\mathbf{A}(G) = f(G) + (Y_1, ..., Y_d) \tag{4}$$

where the Y_i are drawn i.i.d from $Lap(GS_f(G)/\varepsilon)$. The Algorithm satisfies $(\varepsilon, 0)$-differential privacy.

Differential privacy applies equally well to an interactive process, in which an adversary adaptively questions the system about the data. Differential privacy maintains composability, i.e., differential privacy guarantees can be provided even when multiple differentially-private releases are available to an adversary.

Theorem 2. (Composition Theorem[10]) If we have n numbers of (ε, δ)-differentially private mechanisms $M_1, \cdots, M_n$, computed using graph G, then any composition of these mechanisms that yields a new mechanism M is $(n\varepsilon, n\delta)$-differentially private.

Differential privacy can extend to group privacy as well: changing a group of k edges in the data set induces a change of at most a multiplicative $e^{k\epsilon}$ in the corresponding output distribution. In our approach, we generate synthetic graphs by adding controlled perturbations to degree distributions of the original graph and hence in principle we can extend the algorithm to achieve the node differential privacy or the k-edge differential privacy by using the above composition theorem [10]. However, as shown in [12], graph analysis is highly inaccurate under node differential privacy or k-edge differential privacy (when k is large). In our future work, we will investigate the feasibility of graph generation under node differential privacy.

It may be hard to derive the global sensitivity of a complex function or global sensitivity yields unacceptable high noise levels. Nissim et al. [11] introduces a framework that calibrates the instance-specific noise with smaller magnitude than the worst-case noise based on the global sensitivity.

Definition 3. (Local Sensitivity[8, 11]) The local sensitivity of a function $f : D \rightarrow \boldsymbol{R}^d$, ($G \in D$) is

$$LS_f(G) := \max_{G' s.t. G' \in \Gamma(G)} ||f(G) - f(G')||_1. \tag{5}$$

Under the definition of *local sensitivity*, we only consider the set of G' for a given and predetermined G, such that the inclusion or exclusion of a single link between individuals cannot change the output distribution appreciably. We would emphasize that the release $f(G)$ with noise proportional to $LS_f(G)$ cannot achieve rigorous differential privacy as the noise magnitude might reveal information about the database. Refer to Example 1 in [11] for an illustrative example. To satisfy the strict differential privacy, Nissim et al. [11] proposes the β-smooth sensitivity and shows that adding noise proportional to a smooth upper bound on the local sensitivity yields a private output perturbation mechanism.

Definition 4. (Smooth Sensitivity [11]) For $\beta > 0$, the β-sooth sensitivity of $f : D \to \mathbf{R}^d$ ($G \in D$),in the analysis of a given graph G, is

$$S^*_{f,\beta}(G) = \max_{G' \in D} \left(LS_f(G') \cdot e^{-\beta d(G,G')} \right) \tag{6}$$

where $d(G, G')$ is the distance between graphs G and G' (i.e., the number of different edge entries).

Nissim et al. [11] introduces how to compute smooth sensitivity based on the local sensitivity at distance s (measuring how much the sensitivity can change when up to s entries of G are modified).

Definition 5. (Computing Smooth Sensitivity)The sensitivity of f at distance s is

$$LS^{(s)}_f(G) = \max_{G' \in D: d(G,G') \leq s} LS_f(G') \tag{7}$$

The β-smooth sensitivity can be expressed in terms of $LS^{(s)}_f(G)$:

$$\begin{aligned} S^*_{f,\beta}(G) &= \max_{s=0,1,\cdots,n} e^{-s\beta} \left(\max_{G': d(G,G')=s} LS_f(G') \right) \\ &= \max_{s=0,1,\cdots,n} e^{-s\beta} LS^{(s)}(G) \end{aligned} \tag{8}$$

Theorem 3 shows the mechanism of calibrating noise to the smooth upper bound to achieve (ε, δ)-differential privacy. For functions that we cannot compute the smooth sensitivity efficiently or explicitly, Nissim et al. proposes an approximation method that computes the β-smooth upper bound on the local sensitivity of these functions and developed a sample-aggregation framework for a large class of functions [11].

Theorem 3. (Mechanism to Add Noise Based on Smooth Sensitivity[11]) For a function $f : D \to \mathbf{R}^d (G \in D)$, the following mechanism achieves (ϵ, δ)-differential privacy ($\epsilon > 0, \delta \in (0, 1)$):

$$\mathbf{A}(G) = f(G) + \frac{S^*_{f,\beta}(G)}{\alpha} \cdot (Z_1, \cdots, Z_d) \tag{9}$$

where $\alpha = \epsilon/2$, $\beta = \frac{\epsilon}{4(d+ln(2/\delta))}$, and Z_i ($i = 1, \cdots, d$) is drawn i.i.d from $Lap(0, 1)$. Specifically when d=1, β can be reduced as $\beta = \frac{\epsilon}{2ln(2/\delta)}$.

2.2 Graph Generation Models

Over the years, researchers have proposed various graph models to generate graphs that match properties of real networks. Among them, the simplest and most convenient one is the classical E-R random graph $G_{n,p}$[20], which lays the foundation for the typical stochastic approach [19, 21, 22] to generate graphs. With a given expected average degree $\bar{d}$, we can reproduce an n-sized graph by connecting every pair of n nodes with probability $\bar{d}/n$. In this section, we revisit two widely used graph generation models: the dK-graph model [19] and the stochastic Kronecker graph model (SKG) [18].

2.2.1 dK Graph Model

The dK graph model for graph construction mainly applies pseudograph approach, the most common class of graph generation algorithms [23, 24], in constructing graphs matching a desired family of properties called the dK-series in [19]. The dK-series is a finite set of graph properties to describe and constrain random graphs in successively finer detail with the increasing values of d.

The dK-series is defined as the series of properties constraining the generated graph's dK-distribution to be the same form as in a given graph G. dK-distributions are degree correlations within non-isomorphic simple connected subgraphs of size d. For a given graph G, the 0K distribution is simply the average node degree; the 1K distribution is the degree distribution; the 2K-distribution is the joint degree distribution of G which represents the probability that two nodes of degrees k and k' are connected; the 3K-distribution of G is the interconnectivity among triples of nodes. Overall, the dK-series of larger values of d would capture more and more complex detailed properties of the original graph G. In the limit, the dK-distribution describes any given graph completely.

For a given input graph G, the output synthetic 0K-graphs require maintaining the 0K-distribution of G, that is the average node degree; while the output synthetic 1K-graphs reproduce the original graph node degree distribution, and so forth. It is worth pointing out that the degree distribution is different from the degree sequence. The degree sequence is the sequence of length n where each entry $D(i) = d_i$ corresponds to each node's degree whereas the the degree distribution is a distribution vector where each entry $P_1(d_i) = N_{d_i}$ represents the number of nodes whose degree is d_i. Generally, the set of $(d+1)$K-graphs is a subset of dK-graphs. In the whole space of random graphs, the number of possible graphs satisfying the constraints of dK-series would decrease abruptly with the increase of the value of d.

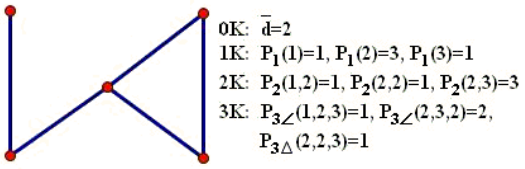


Figure 1: An example of dK-distributions

Figure 1 shows as an example of computing the dK-distributions from a graph G of size 5. For simplicity, the entry value of a dK-distribution is the total number of corresponding d-sized subgraphs. For the given graph G, the 0K-distribution, $P_0 = 2$, is G's average degree; the 1K-distribution, P_1, is G's node degree distribution: $P_1(2) = 3$ means that there are three nodes with degree two; the 2K-distribution, P_2, is the graph G's joint degree distribution; $P_2(2,3) = 3$ means that G contains three edges between 2- and 3-degree nodes; for the 3K-distribution, there are two types of three-sized subgraphs, the triangle and triple that does not form a triangle. In Figure 1, $P_{3\angle}(2,3,2) = 2$ denotes that there are two non-triangle triples where the mid node's degree is three and degrees of the other two nodes are two; while $P_{3\Delta}(2,2,3) = 1$ denotes that there is one triangle formed by three nodes whose degrees respectively are $2, 2, 3$.

The dK-graph model [19] shows surprisingly great performance in capturing global graph structure properties like spectrum and betweenness.

2.2.2 Stochastic Kronecker Graph Model

Kronecker graphs [18] are based on a recursive construction process. The process starts with an initiator graph G_1 with N_1 nodes. By a recursive procedure, larger graphs $G_2, ..., G_n$ are generated. The rth graph generated in the rth recursion, G_r, has about $(N_1)^r$ nodes. Usually, we set $N_1 = 2$. This procedure (Definition 7) is formalized by introducing the concept of Kronecker product (Definition 6) of the adjacency matrices of two graphs.

Definition 6. (Kronecker Product) Given two matrices A and B of size $n \times m$ and $n' \times m'$ respectively, their Kronecker Product is a matrix C of dimensions $(n \cdot n') \times (m \cdot m')$ defined as

$$C = A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1m}B \\ a_{21}B & a_{22}B & \cdots & a_{2m}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}B & a_{n2}B & \cdots & a_{nm}B \end{pmatrix} \tag{10}$$

Definition 7. (Kronecker Power) Given a Kronecker initiator matrix Θ_1, the kth power of Θ_1 is defined by

$$\Theta_1^{[k]} = \Theta_1^{[k-1]} \otimes \Theta_1 = \Theta_1 \otimes \Theta_1 \cdots \otimes \Theta_1 \tag{11}$$

The Stochastic Kronecker graph (SKG) model was proposed in [18]. In the SKG model, each entry of the initiator matrix Θ takes values in the range $[0, 1]$ instead of binary values, representing the probability of that edge being present. Thus following Definition 7 to compute the Kronecker power of Θ_1, each entry of the reproduced stochastic adjacency matrices represents the probability of that particular edge appearing in the corresponding graph. The final synthetic stochastic Kronecker graph is obtained by choosing edge independently with a probability specified by the corresponding entry in the stochastic adjacency matrix(Definition 8).

Definition 8. (Stochastic Kronecker Graphs[18])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 kth 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}.

Applying SKG model to a given graph G, i.e. learning model parameters from G, requires the assumption that G is generated by an SKG model with a specific initiator matrix Θ. Extensive researches [16, 25, 26] have been conducted to studying the problem of the estimation of the true parameter, the initiator matrix Θ for G. In [25], the authors proposed an estimation algorithm which made it possible to enforce differential privacy into SKG generation. Based on their approach, recently, differential privately SKG generation [16] has been achieved by first computing the differentially private degree sequence and the total number of triangles from the original graph, and then using them to compute the private input parameters $\{E, H, T, \Delta\}$ of Moment based estimation [25] which are used to finally generate the private graphs.

3 Private dK-Graph Model

In this section, we respectively propose the approaches to enforcing differential privacy into the 1K- and 2K-distributions based dK-graph generation models. The basic definitions

Table 1: Basic Definitions and Terminologies Used in This Paper

The original graph	G
Number of nodes in G	n
Adjacent matrix of G	A
An entry in A	a_{ij}
The degree of a node i	d_i
The vector of all d_i	D
The degree/1K- distribution	P_1
The 2K-distribution	P_2
The neighboring node set of a node i	$Ngb(i)$
Nodes in $Ngb(i)$ but not in $Ngb(j)$	$Ngb(i) - Ngb(j)$

and terminologies of a graph used in our work are listed in Table 1.

3.1 DP-1K Graph Model

The basic idea of our approach to generate differentially private graphs based on the 1K-graph model is to firstly enforce differential privacy in the calculation of the 1K-distribution and then use the private 1K-distribution as the input of the 1K-graph generator to generate the private 2K-graphs.

In order to enforce differential privacy in the calculation of the 1K-distribution, we firstly give the sensitivity of the 1K-distribution in Claim 1. Based on the global sensitivity, we follow Theorem 1 to add Laplacian noise to the real 1K-distribution computed from the original graph with a given privacy parameters ϵ ($\delta = 0$ in this case). Taking the perturbed 1K-distribution as input, the 1K graph model generator could then generate lots of synthetic graphs while satisfying $(\epsilon, 0)$-differential privacy. In our work, we use the graph model generator software provided in [19]. We conclude this process into Algorithm 1.

Claim 1. The global sensitivity of the 1K degree distribution of a graph G is $GS_{P_1}(G) = 4$.

Proof. When deleting an arbitrary edge (i, j) from graph G, the following four entries of the degree distribution will be changed by exactly one: d_i,d_j,$d_i - 1$,$d_j - 1$. So the global sensitivity is 4. □

Algorithm 1 Private Generation of 1K-graph

Require: Graph G, privacy parameters (ε)
Ensure: $\tilde{G}$ satisfies $(\varepsilon, 0)$-differential privacy
1: Compute the 1K-distribution $P_1(G)$of G
2: Using ε to perturb $P_1(G)$ and acquire ϵ-differentially private $\tilde{P}_1(G)$ //Theorem 1
3: Call procedure *1K_graph_generation*$(\tilde{P}_1(G))$ to generate $\tilde{G}$.

Another possible approach to compute the private 1K-distribution is firstly querying the private degree sequence(the vector containing each node's degree), whose global sensitivity is 2, and then computing the private 1K-distribution from the private degree sequence. However, the degree sequence is a much longer vector than the 1K-distribution; additionally, every entry of the degree sequence vector would have smaller value than that of the 1K degree distribution. Therefore, the degree sequence vector would suffer more from the

relative error though with smaller global sensitivity. With those considerations, we directly query the 1K-distribution from the private graph database server for perturbation.

3.2 DP-2K Graph Model

Algorithm 2 illustrates the detail of our differentially private 2K-graph model, DP-2K. The idea is first computing the differentially private 2K-distribution from the original graph, and then using the private 2K-distribution as the input of the 2K-graph generator to generate the private 2K-graphs.

One challenge here is that the global sensitivity of the 2K-distribution is $4n-7$ (as shown in Claim 2), which is too large to be used for calibrating noise. In [17], the authors explored the use of the local sensitivity to calibrate noise to the 2K-distribution and developed a private 2K-graph model. However, the approach based on the local sensitivity cannot achieve rigorous differential privacy, as shown in [9, 11]. In our algorithm, we use the smooth sensitivity, which is proved to achieve the rigorous differential privacy. We firstly derive the local sensitivity at distance s for the original 2K-distribution $LS^{(s)}_{P_2}(G)$(Claim 2); then compute the smooth sensitivity parameters (β, α) with the given (ϵ, δ) (Line 2) based on Theorem 3; derive the β-smooth sensitivity for P_2 with $\beta, LS^{(s)}_{P_2}(G)$ (Line 3); calibrate noise based on the derived smooth sensitivity and acquire the (ϵ, δ)-differentially private 2K-distribution $\tilde{\tilde{P}}_2$ (Line 4); and finally generate private 2K-graphs (Line 5) with the package provided by [19]. Throug this process, our Algorithm 2 achieves rigorous (ϵ, δ)-differential privacy.

Claim 2. The global sensitivity of the 2K-distribution is $GS_{P_2}(G) = 4n - 7$. The local sensitivity of 2K-distribution is $LS_{P_2}(G) = \max_{i,j\in[n]} 2(d_i + d_j) - 3$. The local sensitivity at distance s of 2K-distribution is $LS^{(s)}_{P_2}(G) = \min\{\max_{i,j\in[n]}\{2(d_i + d_j) - 3 + 2 * s\}, GS_{P_2}(G)\}$

Proof. When deleting an arbitrary edge (i, j) from graph G, the total value change among entries used to involve d_i as one parameter is d_i, for the reason that i leaves the node set of degree d_i; similarly those used to involve d_j will also decrease by d_j in total; specifically, the total amount of decreased value of the above two cases is d_i+d_j-1 since the entry $P_2(d_i, d_j)$ should be only counted once. After deleting edge (i, j), the degree of i, j will be d_i-1, d_j-1, so that the value of entries that used to involve parameter $d_i - 1$ or $d_j - 1$ will be increased by $d_i - 1 + d_j - 1$ in total. So that the local sensitivity is $LS_{P_2}(G) = \max_{i,j\in[n]} 2(d_i + d_j) - 3$. When $d_i = d_j = n - 1$, we have $GS_{P_2}(G) = 4n - 7$. Every time s increase by one, the d_i or d_j will increase by at most one, so that we have $LS^{(s)}_{P_2}(G) = \min\{\max_{i,j\in[n]}\{2(d_i + d_j) - 3 + 2 * s\}, GS_{P_2}(G)\}$. □

Algorithm 2 Private Generation of 2K-graph

Require: Graph G, privacy parameters (ε, δ)
Ensure: $\tilde{G}$ satisfies (ε, δ)-differential privacy
1: Compute the 2K-distribution $P_2(G)$ of G
2: Using the (ε, δ) to compute (β, α) //Theorem 3
3: Compute the β-smooth sensitivity $S^*_{P_2,\beta}(G)$ using $\beta, LS^{(s)}_{P_2}(G)$ //Equation 8
4: Compute $\tilde{P}_2(G)$ using $\alpha, S^*_{P_2,\beta}(G)$ //Equation 9
5: Call procedure $2K\_graph\_generation(\tilde{P}_2(G))$ to generate $\tilde{G}$.

3.3 DP-3K Graph Model

Through in the limit, the dK-series are expected to describe any given graph completely. In principle, we can develop private dK-graph models for varying d. However, when $d >= 3$, the representation of the dK-distribution is complex, which causes the sensitivity (both global sensitivity and smooth sensitivity) significantly large.

Claim 3 shows the sensitivity values of the 3K-distribution. Recall that $P_{3\angle}(d_1, d_2, d_3)$ and $P_{3\Delta}(d_1, d_2, d_3)$ respectively represent the two types of three-sized subgraphs: the triangle and the triple that does not form a triangle. When an arbitrary edge (i, j) is deleted from graph G, many entries in the 3K-distribution will be affected.

Claim 3. The global sensitivity of the 3K-distribution is $GS_{P_3}(G) = \frac{3}{2}(n-2)^2 + 2(n-2)$. The local sensitivity of the 3K-distribution is

$$LS_{P_3}(G) = \max_{i,j\in[n]} \{|S1| + 2|S2| + 2(\sum_{k\in S1}(d_k - 1) - \sum_{k_1,k_2\in T_i, k_1<k_2}(a_{k_1k_2}) - \sum_{k_1,k_2\in T_j, k_1<k_2}(a_{k_1k_2}))\}$$

where $S1 = T_i \cup T_j = \{\{Ngb(i) - Ngb(j)\} \cup \{Ngb(j) - Ngb(i)\}\}$ and $S2 = \{Ngb(i) \cap Ngb(j)\}$. The local sensitivity at distance s of the 3K-distribution is

$$LS^{(s)}_{P_3}(G) = \min\{GS_{P_3}(G), LS_{P_3(G)} + s + 2 \max_{k_q\in S3, t\in i,j} (\sum_{q=1}^{s}(d_{k_q} - \sum_{k_p\in S1t} a_{k_qk_p}) - \sum_{q_1,q_2\in[s];q_1<q_2} a_{k_{q_1}k_{q_2}}(a_{ik_{q_1}}a_{ik_{q_2}} + a_{jk_{q_1}}a_{jk_{q_2}}))\}$$

where $S3 = \{V - \{i, j\} - Ngb(i) - Ngb(j)\}$.

Proof. We use V to denote the vertex set of original graph G; $T_i = Ngb(i) - Ngb(j)$ denotes the set of nodes i's neighbor excluding those being j's neighbor at the same time; $T_j = Ngb(j) - Ngb(i)$ denotes the set of j's neighbor excluding those being i's neighbor; and $S1 = T_i \cup T_j$ is the set of nodes which are either i's neighbor or j's neighbor but not both; $S2 = Ngb(i) \cap Ngb(j)$ is the set of common neighbors of i and j; $S3$ is the set of nodes which are neither i's neighbor nor j's neighbor.

In the local sensitivity, when edge (i, j) is deleted, there are three cases of change among entries of P_3.

Firstly, some non-triangle triples will no longer form three-sized subgraphs. Each of such triple involves i,j and one node is $S1$. They are used to be counted in $P_{3<}(d_i, d_j, d_k)$ ($k \in S1$). There are $|S1|$ of them , causing P_3 changed by $|S1|$.

Secondly, some triangle triples will become non-triangle triples. Such triangle is formed with i,j and one of their shared neighbors. They used to be counted in $P_{3\Delta}(d_i, d_j, d_k)$ ($k \in S2$). After deleting (i,j), they are counted in $P_{3<}(d_i - 1, d_j - 1, d_k)(k \in S2)$. There are $|S2|$ of them , causing P_3 change by $2|S2|$.

Thirdly, some triples(no matter it is triangle or not) involve only one of i and j. The entries of P_3 counting them are changed since i and j jumps from the sets of respectively d_i and d_j to those of$d_i - 1$ and $d_j - 1$. The rest part of $LS^{(s)}_{P_3}(G)$ describes this amount.

Therefore we derive the local sensitivity in the form above.When the local sensitivity gets to its maximum,i.e., $d_i = d_j = \frac{1}{2}(n-2)+1$, $|S2| = 0$, every pair of nodes in $S1$ is connected by an edge, we have

$$\begin{aligned} GS_{P_3}(G) &= 2 \times (n-2)(n-1-1) - 2C^2_{\frac{n-2}{2}} + 0 + (n-2) \\ &= \frac{3}{2}(n-2)^2 + 2(n-2). \end{aligned}$$

For the local sensitivity at distance s, every time s increases by one, we choose one node k_q from $S3$ and add one edge to connect k_q to i or j, it will cause the change of the first case by one, and that of the second case by zero, and that of the third case by two times of the number new three-sized subgraphs brought in by k_q. Thus we have the above form of the smooth sensitivity at distance s. □

Another challenge is that there is no known algorithm to generate dK-graphs for $d >= 3$ given a dK-distribution. The authors in [19] developed an algorithm, the 3K-rewire, for generating 3K-graphs. However, the idea was to modify the original graph G keeping the 3K-distribution unchanged. For private dP-graph models, we can not use the original graph to rewire since the rewired graph may contain other private information than those captured by the dK-distribution. As a result, we only conduct evaluations based on the DP-1K and the DP-2K models.

4 Empirical Evaluation

In this section, we conduct evaluations to compare the three graph generation models: the stochastic Kronecker graph (SKG) model, the 1K-graph model, and the 2K-graph model. For the SKG, we use Gleich's [25] and SNAP library [27]'s codes to generate the synthetic graphs with real parameters learned from the original graphs. For both 1K- and 2K-graph models, we use codes provided by [17] for dK-graph generation. We also implemented our private dK-graph models, DP-1K and DP-2K, by following Algorithms 1 and 2 respectively.

We conduct experiments on four graphs: $CA-GrQC$ (denoted as GC), AS20, Enron, and Polbooks. GC is a co-authorship network from arXiv with 5242 nodes and 14484 edges; AS20 is a technological infrastructure network with 6474 nodes and 12572 edges; these two datasets can be downloaded from SNAP[1]. Enron[2] is an email network collected and prepared by the CALO Project and it has 148 nodes and 869 edges; and Polbooks[3] is a network of books about US politics published around the time of the 2004 presidential election and sold by Amazon.com and it has 105 nodes and 441 edges.

4.1 Topology Metrics

Various metrics can be used to measure the graph utility. Refer to a survey [28] for details. The used graph metrics are shown in Table 2.

- The nodes number(n), edges number(m) and average degree($\bar{d}$) describe the basic scale of the graphs.

[1] $http://snap.stanford.edu/data/index.html$
[2] $http://www.cs.cmu.edu/\ enron/$
[3] $http://www-personal.umich.edu/\ mejn/netdata/$

Table 2: Scalar graph metrics notations

Metric	Notation
Number of nodes	n
Number of edges	m
Average degree	$\bar{d}$
Assortativity coefficient	r
Average clustering	$\bar{C}$
Average distance	$\bar{l}$
Diameter	D
Largest eigenvalue of adjacency matrix	λ
Number of triangles	Δ
Transitivity	t
Betweenness	b
Modularity	Q

- The asssortativity coefficient(r) describes the tendency that nodes with similar degree are connected to each other. Assortative (disassortative) networks are those where nodes with similar (dissimilar) degrees tend to be tightly interconnected. They are more (less) robust to both random and targeted removals of nodes and links.

- The betweenness (b) is a commonly used measure of centrality, i.e., topological importance, both for nodes and links. It is a weighted sum of the number of shortest paths passing through a given node or link. As such, it estimates the potential traffic load on a node or link, assuming uniformly distributed traffic following shortest paths.

- The average distance (l) and the diameter (l_{max}) describe the separation of nodes, which are important for evaluating the performance of routing algorithms as well as of the speed with which worms spread in a network.

- The largest eigenvalue (λ) of the adjacency matrix describes the spectrum character of the graph topology. Eigenvalues provide tight bounds for a number of critical network characteristics [29, 30, 31] including network resilience and network performance like the maximum traffic throughput of the network.

- The average clustering ($\bar{C}$) is the average cluster coefficients of each nodes. The transitivity (t) and the number of triangles (Δ) give the graph level clustering characteristics of the graph.

- The modularity (Q) is defined as the fraction of all edges that lie within communities minus the expected value of the same quantity in a graph in which the vertices have the same degrees but edges are placed at random without regard to the communities. The modularity captures the goodness of the community structure and a value $Q = 0$ indicates that the community structure is no stronger than would be expected by random chance.

For each graph model, we generate 100 random graphs and choose one with the largest average clustering coefficient $\bar{C}$. This strategy was adopted in previous works (e.g., [17]) since the variation of $\bar{C}$ from randomly generated graphs is often small.

4.2 Evaluation Result

We report all our results for four networks in Tables 3,4,5, and 6. In each table, 'Original' denotes the original graph; 'SKG', '1K', and '2K' respectively denote the graphs generated by the SKG, the 1K-graph model, and the 2K-graph model without privacy protection; 'DP1K(ϵ)'denotes the graph generated by the private DP-1K model with a given ϵ value and 'DP2K(ϵ)' denotes the graph generated by the DP-2K model with a given ϵ value. We choose ϵ from (2000,200,20,2,0.2). For the DP-2K model, we use the same $\delta = 0.01$. We do not include the results for the private SKG model since they can be acquired from [16]. As shown in Section 4.2.1, the SKG model (even without privacy requirement) incurs much larger utility loss than the dK-graph models or the private DP-dK models.

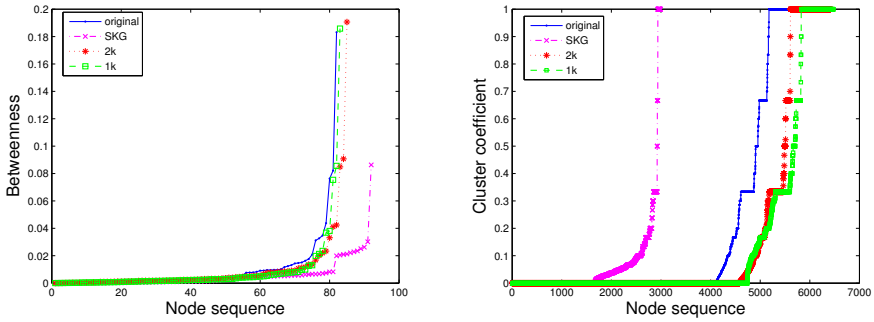
4.2.1 SKG Model VS. dK-Graph Model

Our experiment results show that dK-graph models (both 1K and 2K) outperform the SKG model. The dK-graph models more precisely capture most of the evaluated graph properties than the SKG model. Taking the graph AS20 (6474 nodes,12572 edges) as an example, the 2K-graph model outperforms the SKG model with nine out of the ten metrics used in our evaluation; and the 1K-graph model outperforms the SKG model with seven metrics. The first five columns of Table 4 show the detailed metric values. Compared to the original graph, the relative errors of metrics n, m and d of 1K- and 2K-graph models are around 0.1%, which indicates the dK-graph models can well capture the scale of networks. On the contrary, the SKG model generates graphs that often have different scales than the original one. For example, the relative errors of n, m and d are $54\%, 23\%, 166\%$ for the SKG model. This is because of the limitation of the SKG model that it could only generate graph with node number near 2^r (r is the iterative parameter of the model).

Apart from the global characteristics, the dK-graph models also show better performance in per-node metrics than the SKG model. As the evidence, Figure 2 shows the overlayed patterns of the distribution of the node betweenness (b) and the cluster coefficient sequence (C) of the original graph AS20 as well as its corresponding ones generated by the SKG and dK-graph models. Figure 2(a) plots the sorted node betweenness distribution, where both the two lines representing 1K- and 2K-graph models are much similar and closer to the line representing the original AS20 graph. Figure 2(b) plots the sorted cluster coefficient sequence, where both 1K- and 2K-graph models more accurately reproduce the cluster coefficient sequence even to the positions of every turning point in the line representing the original graph. On the contrary, both the node betweenness distribution and the cluster coefficient sequence from the SKG graph are significantly different from the original graph.

We would also point out that neither the dK-graph models nor the SKG model could accurately capture the average cluster coefficient $\bar{C}$, number of triangles Δ, transitivity t, and the modularity Q where in most cases, the relative errors for them are more than 50%. For example, as shown in the last rows of Tables 3,4,5, and 6, the community structure (in terms of modularity Q) is much lost across all graph generation models due to the nature of random generation. However, we can see that the dK-graph models preserve more community structure than the SKG model. For the assortativity coefficient (r), it could only be precisely captured by the 2K-graph model. We can see from Line 5 of Table 3 that, for graph GC (5242 nodes and 14484 edges), the relative error of r for the 2K-graph is 2.3% while the relative errors for those generated by the SKG and the 1K-graph model are more than 90%.

Finally, we would emphasize that the SKG model cannot achieve utility preservation as



(a) Betweenness Distribution (b) Cluster Coefficient Sequence

Figure 2: Overlayed patterns of real network for AS20 and generated graphs

well as the dK-graph models. Even our private DP-dK models can achieve better utility preservation than the SKG model without privacy enforcement. For instance, as shown in Table 5 for the Enron graph, the DP-1K model with strong privacy protection ($\epsilon = 2$) outperforms the SKG model in terms of utility preservation with eight metrics. The DP-2K graph model with weak privacy (satisfying $(200, 0.01)$-differential privacy) also outperforms the SKG model with eight metrics. In summary, our evaluation demonstrates that the dK-graph models (both 1K and 2K), even with acceptable amount of perturbation, would generate graphs with better utility than the SKG model.

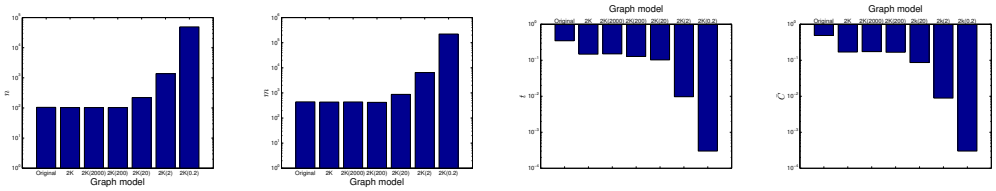
4.2.2 Privacy vs. Utility of DP-dK Model

In this section, we focus on the tradeoff between utility and privacy of our private dK-graph models. As shown in Section 3.2, the 2K-distribution often has the large sensitivity. When enforcing strong privacy protection with small ϵ values like 0.2 or 2, the DP-2K graph model would incur significant utility loss. For example, the generated graphs often have extremely large scale and uncertain values of other graph metrics. However, for weak or no privacy protection, the private DP-2K graph model outperforms the DP-1K graph model, i.e., capturing more information of the original graph, especially for the assortativity coefficient (r), the average cluster coefficient ($\bar{C}$), and the modularity (Q); for other metrics like $n, m, \bar{d}, \lambda, l, D, t$, the DP-2K model shows at least the same level of accuracy as the DP-1K model.

Figure 3 shows the values of metrics $n, m, t, \bar{C}$ of the Polbooks graph and its corresponding graphs generated by the DP-2K model with varying ϵ values. Each pillar from left to right corresponds to the original graph, the 2K-graph model, and the DP-2K models with varying ϵ values from 2000 to 0.2 respectively. We observe that the magnitude of metric values changes dramatically as ϵ decreases from 20 to 0.2 in all the four metrics, which indicate the significant utility loss for the DP-2K model with strong privacy enforcement. Contrastively, Figure 4 show values for the 1K-model and the DP-1K models with varying ϵ values. We can see that the DP-1K model well preserves the utility even with small ϵ values like 2 and 0.2. We can also observe from Figures 3 and 4 that the DP-2K model achieves better utility preservation than the DP-1K model under the weak privacy enforcement. For example, when $\epsilon = 200, 2000$, the DP-2K model has more accurate $\bar{C}$ than the DP-1K model (no difference for n, m, t).

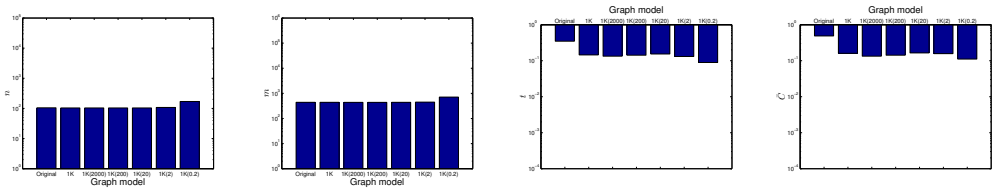
Additionally, our experiment results show that the assortativity coefficient (r) can only be precisely achieved by the DP-2K model. For example, as shown in Line 5 of Table 3 for the graph GC, the DP-2K graph model with $(200, 0.01)$-differential privacy incurs much smaller loss (with the relative error of 7.8%) than both the 1K-graph model and the DP-1K graph models (with the relative errors more than 90%). The assortativity coefficient is an important metric to describe the tendency that nodes with similar degree are connected.

To sum up, our evaluations show that the DP-2K graph model generally achieves better utility than the DP-1K graph model for large ϵ values whereas the DP-1K graph model would achieve better utility for small ϵ values. We also would point out that large ϵ values (e.g., (2000,200,20) provide almost no privacy protection in practice and our evaluation with large ϵ values is to check the performance of the DP-2K graph model in real social networks. As shown in Tables 3 and 4, the DP-2K model is infeasible in terms of privacy protection for GC and AS20 graphs as the smallest ϵ value is already 20. However, as shown in Tables 5 and 6, the DP-2K model outperforms the DP-1K model for Enron and Polbooks graphs even with small ϵ values.



(a) Number of Nodes (b) Number of Edges (c) Transitivity (d) Average Node Clustering

Figure 3: Utility change with varying ϵ on the DP-2K private model generated graphs for Polbooks



(a) Number of Nodes (b) Number of Edges (c) Transitivity (d) Average Node Clustering

Figure 4: Utility change with varying ϵ on the DP-1K private model generated graphs for Polbooks

5 Conclusion and Future Work

In this paper, we have developed private dK-graph generation models that enforce rigorous differential privacy while preserving utility. We have conducted theoretical analysis and empirical evaluations to show that the developed private dK-graph generation models significantly outperform the approach based on the stochastic Kronecker generation model. We have shown that the DP-2K graph model generally achieves better utility preservation

Table 3: Metrics Evaluation of Graph GC

	Original	SKG	1K	2K	DP1K (ϵ)			DP2K (ϵ)		
					20	2	0.2	2000	200	20
n	5241	3522	5241	4581	5242	5239	5382	4585	4652	6106
m	14484	13767	14484	12748	14509	14596	19430	12724	12949	24415
$\bar{d}$	5.527	7.817	5.527	5.565	5.535	5.572	7.220	5.550	5.567	7.997
r	0.659	0.016	0.017	0.643	-0.018	-0.007	-0.005	0.645	0.608	0.427
$\bar{C}$	0.529	0.014	0.008	0.018	0.007	0.008	0.015	0.017	0.015	0.011
$\bar{l}$	3.807	3.789	4.269	4.314	4.220	4.226	3.849	4.314	4.279	4.076
D	17	9	11	14	13	12	10	15	16	19
λ	45.616	23.029	18.259	41.039	17.594	18.182	27.770	40.460	34.347	38.785
Δ	48260	1585	775	17650	628	745	3035	17149	12303	12485
t	0.629	0.017	0.010	0.269	0.008	0.009	0.017	0.264	0.192	0.064
Q	0.801	0.317	0.407	0.503	0.404	0.402	0.323	0.506	0.495	0.401

Table 4: Metrics Evaluation of Graph AS20

	Original	SKG	1K	2K	DP1K (ϵ)			DP2K (ϵ)		
					20	2	0.2	2000	200	20
n	6474	2987	6474	6418	6475	6491	7006	6422	9678	49788
m	12572	15456	12571	12450	12585	15705	38431	12863	30716	240675
$\bar{d}$	3.883	10.348	3.883	3.879	3.887	4.839	10.970	4.005	6.347	9.667
r	-0.181	-0.176	-0.173	-0.182	-0.173	-0.324	-0.579	-0.175	-0.113	-0.062
$\bar{C}$	0.252	0.081	0.149	0.164	0.148	0.338	0.597	0.113	0.056	0.010
$\bar{l}$	3.705	3.050	3.221	3.448	3.277	2.921	2.733	3.548	3.705	4.027
D	9	6	12	8	16	9	7	9	8	11
λ	46.31	39.60	49.89	42.81	50.39	70.79	161.75	41.14	51.43	–
Δ	6584	7052	11732	4373	12143	42351	549789	3087	12830	–
t	0.009	0.027	0.017	0.006	0.017	0.030	0.103	0.004	0.012	–
Q	0.608	0.250	0.480	0.513	0.478	0.402	0.230	0.505	0.385	0.359

Table 5: Metrics Evaluation of Graph Enron

	Original	SKG	1K	2K	DP1K (ϵ)			DP2K (ϵ)		
					20	2	0.2	20	2	0.2
n	148	254	148	146	147	153	281	582	6273	106976
m	869	1804	868	843	867	1024	1538	3024	29090	512785
$\bar{d}$	11.74	14.31	11.73	11.54	11.79	13.38	10.94	10.39	9.27	9.58
r	-0.146	-0.223	-0.062	-0.148	-0.083	-0.077	-0.050	-0.062	-0.176	-0.019
$\bar{C}$	0.512	0.219	0.189	0.199	0.195	0.242	0.092	0.053	0.003	0.002
$\bar{l}$	2.514	2.296	2.295	2.266	2.280	2.240	2.614	3.007	3.992	4.997
D	6	4	4	4	4	5	5	6	8	8
λ	17.83	22.98	17.79	16.99	17.64	21.71	16.04	19.09	19.02	20.52
Δ	1700	1687	821	684	767	1519	599	845	652	–
t	0.344	0.124	0.167	0.146	0.155	0.210	0.078	0.048	0.004	–
Q	0.417	0.198	0.224	0.233	0.216	0.189	0.256	0.313	0.329	0.278

Table 6: Metrics Evaluation of Graph Polbooks

	Original	SKG	1K	2K	DP1K (ϵ)			DP2K (ϵ)		
					20	2	0.2	20	2	0.2
n	105	128	104	103	104	108	170	221	1374	49089
m	441	849	440	433	440	448	712	873	6459	220939
$\bar{d}$	8.40	13.163	8.442	8.408	8.462	8.296	8.377	8.275	9.402	9.002
r	-0.128	-0.106	-0.094	-0.129	-0.028	-0.108	0.025	-0.023	0.029	-0.008
$\bar{C}$	0.487	0.177	0.160	0.169	0.164	0.157	0.111	0.086	0.008	0.0003
$\bar{l}$	3.078	2.132	2.383	2.384	2.409	2.398	2.638	2.775	3.557	4.971
D	7	4	4	4	4	4	5	5	8	12
λ	11.93	18.01	11.87	11.59	12.17	11.58	12.56	12.75	13.30	13.58
Δ	560	825	231	230	246	208	239	318	243	221
t	0.348	0.176	0.144	0.147	0.153	0.130	0.089	0.102	0.009	0.0003
Q	0.502	0.199	0.269	0.286	0.273	0.270	0.283	0.336	0.350	0.285

than the DP-1K graph model with weak privacy enforcement (very large ϵ value) whereas the DP-1K graph model would achieve better utility preservation with strong privacy enforcement (small ϵ value).

There are some other aspects of this work that merit further research. Among them, We are interested in how to preserve some known graph metrics in addition to degree correlations in the dK-graph generation process. We will continue the line of this research by investigating how to enforce edge differential privacy on other graph generation models (e.g., the class of exponential random graph models [32]) and comparing various models in terms of the tradeoff between utility and privacy. We will explore whether it is feasible to enforce node differential privacy or k-edge differential privacy on existing graph generation models. Most recently, ρ-differential identification [33] was proposed to restrict the probability of individual re-identification under a parameter ρ. We will explore how our graph generation models work under ρ-differential identification.

Acknowledgment

The authors would like to thank anonymous reviewers for their valuable comments and suggestions. This work was supported in part by U.S. National Science Foundation (CCF-1047621, CNS-0831204, IIS-0546027) and U.S. National Institute of Health (1R01GM103309).

References

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

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

[3] X. Ying and X. Wu, "Graph generation with prescribed feature constraints," in *SDM*, 2009.

[4] K. Liu and E. Terzi, "Towards identity anonymization on graphs," in *SIGMOD*, 2008.

[5] B. Zhou and J. Pei, "Preserving privacy in social networks against neighborhood attacks," in *ICDE*, Cancn, Mxico, 2008.

[6] L. Zou, L. Chen, and M. T. Özsu, "K-automorphism: A general framework for privacy preserving network publication," in *VLDB*, 2009.

[7] X. Ying and X. Wu, "On link privacy in randomizing social networks," in *PAKDD*, 2009.

[8] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," *Theory of Cryptography*, pp. 265–284, 2006.

[9] C. Dwork, "A firm foundation for private data analysis," *Communications of the ACM*, vol. 54, no. 1, pp. 86–95, 2011.

[10] C. Dwork and J. Lei, "Differential privacy and robust statistics," in *STOC*, 2009, pp. 371–380.

[11] K. Nissim, S. Raskhodnikova, and A. Smith, "Smooth sensitivity and sampling in private data analysis," in *STOC*, 2007, pp. 75–84.

[12] M. Hay, C. Li, G. Miklau, and D. Jensen, "Accurate estimation of the degree distribution of private networks," in *ICDM*, 2009, pp. 169–178.

[13] V. Rastogi, M. Hay, G. Miklau, and D. Suciu, "Relationship privacy: Output perturbation for queries with joins," in *PODS*, 2009, pp. 107–116.

[14] Y. Wang, X. Wu, J. Zhu, and Y. Xiang, "On learning cluster coefficient of private networks," in *ASONAM*, 2012.

[15] Y. Wang, X. Wu, and L. Wu, "Differential privacy preserving spectral graph analysis," in *PAKDD*, 2013.

[16] D. Mir and R. Wright, "A differentially private graph estimator," in *ICDMW*, 2009, pp. 122–129.

[17] A. Sala, X. Zhao, C. Wilson, H. Zheng, and B. Zhao, "Sharing graphs using differentially private graph models," in *SIGCOM*, 2011, pp. 81–98.

[18] J. Leskovec and C. Faloutsos, "Scalable modeling of real graphs using kronecker multiplication," in *ICML*, 2007, pp. 497–504.

[19] P. Mahadevan, D. Krioukov, K. Fall, and A. Vahdat, "Systematic topology analysis and generation using degree correlations," in *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4.ACM, 2006, pp. 135–146.

[20] P. ERDdS and A. R&WI, "On random graphs i." *Publ. Math. Debrecen*, vol. 6, pp. 290–297, 1959.

[21] M. Boguñá and R. Pastor-Satorras, "Class of correlated random networks with hidden variables," *Physical Review E*, vol. 68, no. 3, p. 036112, 2003.

[22] F. Chung and L. Lu, "Connected components in random graphs with given expected degree sequences," *Annals of combinatorics*, vol. 6, no. 2, pp. 125–145, 2002.

[23] W. Aiello, F. Chung, and L. Lu, "A random graph model for massive graphs," in *STOC*, 2000, pp. 171–180.

[24] M. Molloy and B. Reed, "A critical point for random graphs with a given degree sequence," *Random Structures & Algorithms*, vol. 6, no. 2-3, pp. 161–180, 1995.

[25] D. Gleich and A. Owen, "Moment based estimation of stochastic kronecker graph parameters," *Internet Math*, vol. 8, no. 3, pp. 232–256, 2012.

[26] J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani, "Kronecker graphs: An approach to modeling networks," *The Journal of Machine Learning Research*, vol. 11, pp. 985–1042, 2010.

[27] J.Leskovec., "Snap: Stanford network analysis platform."

[28] L. da F. Costa, F. A. Rodrigues, G. Travieso, and P. R. V. Boas, "Characterization of complex networks: A survey of measurements," *Advances In Physics*, vol. 56, p. 167, 2007.

[29] F. Chung, *Spectral graph theory*. Amer Mathematical Society, 1997, no. 92.

[30] L. Li, D. Alderson, W. Willinger, and J. Doyle, "A first-principles approach to understanding the internet's router-level topology," in *ACM SIGCOMM Computer Communication Review*, vol. 34,

no. 4. ACM, 2004, pp. 3–14.

[31] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger, "Network topology generators: Degree-based vs. structural," in *ACM SIGCOMM Computer Communication Review*, vol. 32, no. 4. ACM, 2002, pp. 147–159.

[32] C. Anderson, S. Wasserman, and B. Crouch, "A p* primer: Logit models for social networks," *Social Networks*, vol. 21, no. 1, pp. 37–66, 1999.

[33] J. Lee and C. Clifton, "Differential identifiability," in *KDD*, 2012, pp. 1041–1049.