Sharing Graphs using Differentially Private Graph Models

Alessandra Sala
Computer Science Dept.
UC Santa Barbara
Santa Barbara, CA 93106
alessandra@cs.ucsb.edu

Xiaohan Zhao
Computer Science Dept.
UC Santa Barbara
Santa Barbara, CA 93106
xiaohanzhao@cs.ucsb.edu

Christo Wilson
Computer Science Dept.
UC Santa Barbara
Santa Barbara, CA 93106
bowlin@cs.ucsb.edu

Haitao Zheng
Computer Science Dept.
UC Santa Barbara
Santa Barbara, CA 93106
htzheng@cs.ucsb.edu

Ben Y. Zhao
Computer Science Dept.
UC Santa Barbara
Santa Barbara, CA 93106
ravenben@cs.ucsb.edu

ABSTRACT

Continuing success of research on social and computer networks requires open access to realistic measurement datasets. While these datasets can be shared, generally in the form of social or Internet graphs, doing so often risks exposing sensitive user data to the public. Unfortunately, current techniques to improve privacy on graphs only target specific attacks, and have been proven to be vulnerable against powerful de-anonymization attacks.

Our work seeks a solution to share meaningful graph datasets while preserving privacy. We observe a clear tension between strength of privacy protection and maintaining structural similarity to the original graph. To navigate the tradeoff, we develop a *differentially-private graph model* we call Pygmalion. Given a graph G and a desired level of ϵ-differential privacy guarantee, Pygmalion extracts a graph's detailed structure into degree correlation statistics, introduces noise into the resulting dataset, and generates a synthetic graph G'. G' maintains as much structural similarity to G as possible, while introducing enough differences to provide the desired privacy guarantee. We show that simply applying differential privacy to graphs results in the addition of significant noise that may disrupt graph structure, making it unsuitable for experimental study. Instead, we introduce a partitioning approach that provides identical privacy guarantees using much less noise. Applied to real graphs, this technique requires an order of magnitude less noise for the same privacy guarantees. Finally, we apply our graph model to Internet, web, and Facebook social graphs, and show that it produces synthetic graphs that closely match the originals in both graph structure metrics and behavior in application-level tests.

Categories and Subject Descriptors

H.3.5 [**Information Storage and Retrieval**]: Online Information Services—*Data sharing*; K.4.1 [**Computers and Society**]: Public Policy Issues—*Privacy*

General Terms

Algorithms, Security

Keywords

Differential Privacy, Graph Models, Online Social Networks

IMC'11, November 2–4, 2011, Berlin, Germany.

1. INTRODUCTION

Studying structure of real social and computer networks through graph analysis can produce insights on fundamental processes such as information dissemination, viral spread and epidemics, network dynamics and resilience to attacks [4,26,27,38]. The use of real graphs generated from measurement data is invaluable, and can be used to validate theoretical models or realistically predict the effectiveness of applications and protocols [2,12,41,43].

Unfortunately, there is often a direct tension between the need to distribute real network graphs to the research community, and the privacy concerns of users or entities described by the dataset. For example, social graphs from real measurements are used to capture a variety of artifacts in online social networks, including strength of social ties, number and frequency of social interactions, and flow of information. Similarly, detailed topology graphs of enterprise networks or major ISPs contain confidential information about the performance and robustness of these networks. Releasing such sensitive datasets for research has been challenging. Despite the best of intentions, researchers often inadvertently release more data than they originally intended [35,36,47]. Past experience has taught us that traditional anonymization techniques provide limited protection, and often can be overcome by privacy attacks that "de-anonymize" datasets using external or public datasets [5,35,36].

Thus we are left asking the question, *how can researchers safely share realistic graph datasets from measurements without compromising privacy?* One option is to develop and apply stronger anonymization techniques [24,30], many of which modify the graph structure in subtle ways that improve privacy but retain much of the original graph structure. However, these approaches generally only provide resistance against a specific type of attack, and cannot provide protection against newly developed deanonymization techniques. Techniques exist in the context of databases and data mining which provide provable levels of protection [18,19], but are not easily applied to graphs. Still other techniques can protect privacy on graphs, but must significantly change the graph structure in the process [24,39].

Our approach to provide graph privacy and preserve graph structure. We seek a solution to address the above question, by starting with observation that any system for sharing graphs must deal with the tension between two goals: *protecting privacy* and *achieving structural similarity to the original, unmodified graph*. At one extreme, we can distribute graphs that are isomorphic to the original, but vulnerable to basic deanonymization attacks. At the other extreme, we can distribute random graphs that share no structural similarities to the original. These graphs will not yield any meaningful information to privacy attacks, but they are also not useful to researchers, because they share none of the real structures of the original graph.

Ideally, we want a system that can produce graphs that span the entire privacy versus similarity spectrum. In such a system, users can specify a desired level of privacy guarantee, and get back a set of graphs that are similar to the real graph in structure, but have enough differences to provide the requested level of privacy.

The main premise of our work is that we can build such a system, by distilling an original graph G into a statistical representation of graph structure, adding controlled levels of "noise," and then generating a new graph G' using the result statistics. This requires two key components. First, we need a way to accurately capture a graph's structure as a set of structural statistics, along with a generator that converts it back into a graph. For this, we use the *dK-series*, a graph model that is capable of capturing sufficient graph structure at multiple granularities to uniquely identify a graph [13, 31]. We can achieve the desired level of privacy by introducing a specific level of noise into G's degree correlation statistics. Second, we need a way to determine the appropriate noise necessary to guarantee a desired level of privacy. For this, we develop new techniques rooted in the concept of ϵ*-differential privacy*, a technique previously used to quantify privacy in the context of statistical databases.

In this paper, we develop *Pygmalion*, a differentially private graph model for generating synthetic graphs. Pygmalion preserves as much of the original graph structure as possible, while injecting enough structural noise to guarantee a chosen level of privacy against privacy attacks. Initially, we formulate a basic differentially private graph model, which integrates controlled noise into the dK degree distributions of an original graph. We use the dK-2 series, which captures the frequency of adjacent node pairs with different degree combinations as a sequence of frequency values. However, when we derive the necessary conditions required to achieve ϵ-differential privacy, they show that an asymptotical bound for the required noise grows polynomially with the maximum degree in the graph. Given the impact of dK values on graph structure, these large noise values result in synthetic graphs that bear little resemblance to the original graph.

To solve this challenge, we seek a more accurate graph model by significantly reducing the noise required to obtain ϵ-differential privacy. We develop an algorithm to partition the statistical representation of the graph into clusters, and prove that by achieving ϵ-differential privacy in each cluster, we achieve the same property over the entire dataset. Using a degree-based clustering algorithm, we reduce the variance of degree values in each cluster, thereby dramatically reducing the noise necessary for ϵ-differential privacy. Finally, we apply isotonic regression [6] as a final optimization to further reduce the effective error by more evenly distributing the added noise.

We apply our models to a number of Internet and Facebook graphs ranging from 14K nodes to 1.7 million nodes. The results show that for a given level of privacy, our degree-based clustering algorithm reduces the necessary noise level by *one order of magnitude*. Isotonic regression further reduces the observed error in dK values on our graphs by 50%. Finally, we experimentally show that for moderate privacy guarantees, synthetic graphs generated by Pygmalion closely match the original graph in both standard graph metrics and application-level experiments.

Access to realistic graph datasets is critical to continuing research in both social and computer networks. Our work shows that differentially-private graph models are feasible, and Pygmalion is a first step towards graph sharing systems that provide strong privacy protection while preserving graph structures.

2. GRAPHS AND DIFFERENTIAL PRIVACY

In this section, we provide background on graph anonymization techniques, and motivate the basic design of our approach to graph anonymization. First, we discuss prior work, the inherent challenges in performing graph anonymization, and our desired privacy goals. Second, we introduce the main concepts of ϵ-Differential Privacy, and lay out the preconditions and challenges in leveraging this technique to anonymize graphs. Finally, we motivate the selection of the dK-series as the appropriate graph model on which to build our system.

2.1 Data Privacy: Background and Goals

A significant amount of prior work has been done on protecting privacy of datasets. We summarize them here, and clarify our privacy goals in this project.

Private Datasets. Many research efforts have developed privacy mechanisms to secure large datasets. Most of these techniques, including cryptographic approaches [7] and statistical perturbations [19, 37], are designed to protect structured data such as relational databases, and are not applicable to graph datasets. An alternative, probabilistic approach to privacy is k-anonymity [42]. It is designed to secure sensitive entries in a table by modifying the table such that each row has at least $k-1$ other rows that are identical [18]. Several public datasets have been successfully anonymized with k-anonymity [1, 33] or through clustering-based anonymization strategies [8].

Graph Anonymization. Several graph anonymization techniques have been proposed to enable public release of graphs without compromising user privacy. Generally, these techniques only protect against specific, known attacks. The primary goal of these anonymization techniques is to prevent attackers from identifying a user or a link between users based on the graph structure. Several anonymization techniques [24, 30, 39, 46, 48] leverage the k-anonymity model to create either k identical neighborhoods, or k identical-degree nodes in a target graph. These types of "attack-specific" defenses have two significant limitations. First, recent results have repeatedly demonstrated that researchers or attackers can invent novel, unanticipated de-anonymization attacks that destroy previously established privacy guarantees [5, 35, 36, 45]. Second, many of these defenses require modifications to the protected graph that significantly alter its structure in detectable and meaningful ways [24, 39].

Our Goals: Edge vs. Node Privacy. In the context of privacy for graphs, we can choose to focus on protecting the privacy of either node or edges. As will become clear later in this paper, our approach of using degree correlations (*i.e.* the dK-series), captures graph structure in terms of different subgraph sizes, ranging from 2 nodes connected by a single edge (dK-2) to larger subgraphs of size K.

Our general approach is to produce synthetic graphs by adding controlled perturbations to the graph structure of the original graph.

This approach can provide protection for both node privacy and edge privacy. This choice directly impacts the sensitivity of the graph privacy function, and as a result, how much structural noise must be introduced to obtain a given level of privacy guarantees.

In this paper, we choose to focus on edge privacy as our goal, and apply this assumption in our analysis of our differential privacy system in Section 3. We chose to target edge privacy because our work was originally motivated by privacy concerns in sharing social graphs, where providing edge privacy would address a number of practical privacy attacks.

2.2 Differential Privacy

Our goal is to create a novel system for the generation of anonymized graphs that support two key properties:

1. Provides quantifiable privacy guarantees for graph data that are "future-proof" against novel attacks.
2. Preserves as much original graph structure as possible, to ensure that anonymized data is still useful to researchers.

Differential privacy [14] is a recently developed technique designed to provide and quantify privacy guarantees in the context of statistical databases [15,25]. Others have demonstrated the versatility of this technique by applying differential privacy to distributed systems [40], network trace anonymization [32], data compression techniques [44], and discrete optimization algorithms [22]. Other work focused specifically on applying differential privacy to simple graph structures such as degree distributions [23, 25]. In contrast, our work has the potential to inject changes at different granularities of substructures in the graph, instead of focusing on a single graph metric.

One piece of prior work tried to guarantee graph privacy by adding differential privacy to Kronecker graphs [34]. Whereas this approach tries to guarantee privacy by perturbing the Kronecker model parameters, our strategy acts directly on graph structures, which provides tighter control over the perturbation process. Unfortunately, the author asserts there are incorrect results in the paper[1].

Basic Differential Privacy. The core privacy properties in differential privacy are derived from the ability to produce a query output Q from a database D, which could also have been produced from a slightly different database D', referred to as D's neighbor [14].

DEFINITION 1. *Given a database D, its neighbor database D' differs from D in only one element.*

We obtain differential privacy guarantees by injecting a controlled level of statistical noise into D [16]. The injected noise is calibrated based on the sensitivity of the query that is being executed, as well as the statistical properties of the Laplace stochastic process [17]. The *sensitivity* of a query is quantified as the maximum amount of change to the query's output when one database element is modified, added, or removed. Together, query sensitivity and the ϵ value determine the amount of noise that must be injected into the query output in order to provide ϵ-differential privacy.

Differential privacy works best with *insensitive queries*, since higher sensitivity means more noise must be introduced to attain a given desired level of privacy. Thus insensitive queries introduce lower levels of errors, and provide more accurate query results.

[1]See the author's homepage.

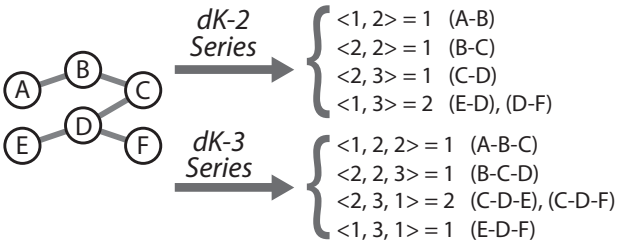


Figure 1: An illustrative example of the dK-series. The dK-2 series captures the number of 2-node subgraphs with a specific combination of node-degrees, and the dK-3 captures the number of 3-node subgraphs with distinct node-degree combinations.

2.3 Differential Privacy on Graphs

We face two key challenges in applying differential privacy concepts to privacy protection on graphs. First, we must determine a "query" function in our context which we can use to apply differential privacy concepts. Second, the sensitivity of this query function must be low enough, so that we can attain privacy guarantees by introducing only low levels of noise, thus allowing us to preserve the accuracy of the results. In our context, this means that we want to generate graphs that retain the structure and salient properties of the original graph. We address the former question in this section by proposing the use of the dK-series as our graph query operation. We address the accuracy question in Sections 3 and 4, after fully explaining the details of our system.

Recall that the problem we seek to address is to anonymize graph datasets so that they can be safely distributed amongst the research community. We leverage a *non-interactive* query model [14], such that the original graph structure is queried only once and the entire budget to enforce privacy is used at this time. dK is used to query the graph and the resulting dK-series is perturbed under the differential privacy framework. Note that only the differentially private dK-series is publicized. Unlike applications of differential privacy in other contexts, we can now generate multiple graphs using this differentially private dK-series without disrupting the level of privacy of the original graph. Therefore, we use a non-interactive query model to safely distributed graph datasets without being constrained to a single dataset.

The dK-Graph Model. We observe that the requirements of this query function can be met by a descriptive graph model that can transform a graph into a set of structural statistics, which are then used to generate a graph with structure similar to the original. Specifically, we propose to use the dK-graph model [31] and its statistical series as our query function. dK captures the structure of a graph at different levels of detail into statistics called dK-series. dK can analyze an original graph to produce a corresponding dK-series, then use a matching generator to output a synthetic graph using the dK-series values as input. The dK-series is the degree distribution of connected components of some size K within a target graph. For example, dK-1 captures the number of nodes with each degree value, *i.e.* the node degree distribution. dK-2 captures the number of 2-node subgraphs with different combinations of node degrees, *i.e.* the joint degree distribution. dK-3 captures the number of 3-node subgraphs with different node degree combinations, *i.e.* an alternative representation of the clustering coefficient distribution. dK-n (where n is the number of nodes in the graph) captures the complete graph structure. We show a detailed

example in Figure 1, where we list dK-2 and dK-3 distributions for a graph.

dK is ideal for us because the dK-series is a set of data tuples that provides a natural fit for injecting statistical noise to attain differential privacy. In addition, together with their matching generators, higher levels of dK-series, *i.e.* $n > 3$, could potentially provide us with a bidirectional transformation from a graph to its statistical representation and back.

While larger values of K will capture more structural information and produce higher fidelity synthetic graphs, it comes at the expense of higher computation and storage overheads. Our work focuses on the dK-2 series, because generator algorithms have not yet been discovered for dK-series where $K \geq 3$. While this may limit the accuracy of our current model, our methodology is general, and can be used with higher order dK-series when their generators are discovered.

ϵ-Differential Privacy in Graphs. Given the above, we can now outline how to integrate differential privacy in the context of graphs. An ϵ-differentially private graph system would output a graph that given a statistical description of an input graph, the probability of seeing two similar graphs as the real input graph is close, where closeness between the two probabilities is quantified by ϵ. A larger value of ϵ means it is easier to identify the source of the graph structure, which means a lower level of graph privacy.

Prior work has demonstrated that in many cases, accuracy of query results on differentially private databases can be improved by decomposing complex queries into sequences of "simple counting queries" that happen to have extremely low sensitivity [9, 10, 15]. Unfortunately, this approach will not work in our context, since our goal is to achieve privacy guarantees on whole graph datasets, and not just privacy for simple graph queries such as node degree distributions. In the next section, we start with a basic formulation of a differentially private graph model, and then provide an optimized version. We illustrate the final process, shown as *Pygmalion* in Figure 2.

3. FIRST STEPS

In this section, we perform the analytical steps necessary to integrate ϵ-differential privacy into the dK graph model. Our goal is to derive the amount of noise necessary to achieve a given ϵ-privacy level. The amount of Laplacian noise necessary is a function of both ϵ, the user-specified privacy parameter, and S, the sensitivity of the dK function. First, we formally define the dK-2 series, and derive its sensitivity S_{dK-2}. Next, we describe the dK-perturbation algorithm (dK-PA) for injecting noise into the original dK-2 series, and prove that it provides the desired ϵ-differential privacy. Our analysis shows that the asymptotic bound on noise used in dK-PA grows polynomially with maximum node degree, which means we need to inject relatively large levels of noise to guarantee ϵ-privacy. Finally, as expected, our experiments on real graphs confirm that dK-PA generates synthetic graphs with significant loss in accuracy. This poor result motivates our search for improved techniques in Section 4.

3.1 Sensitivity of dK-2

dK-function. We formally define dK-2 as a function over a graph $G = (V, E)$, where V is the set of nodes and E is the set of edges connecting pair of nodes in V:

$$dK(G) : G^n \rightarrow \Im$$

where G^n is the set of graphs with $n = |V|$ nodes and $\Im$ is the set of unique degree tuples in the dK-2-series with the corresponding count of instances in G. Formally, $\Im$ is a collection of $\{d_x, d_y; k\}$ where each entry represents that the number of connected components of size 2 with degree (d_x, d_y) is k. Let m be the cardinality of $\Im$. Because the maximum number of entries in dK-2 is bounded by the number of possible degree pairs, $\sum_{i=1}^{d_{max}} i$, where d_{max} be the maximum node degree in G, thus $m = O(d_{max}^2)$. Prior studies have demonstrated that in large network graphs d_{max} is upper bounded by $O(\sqrt{n})$ [29,43], and thus, in those cases, m is upper bounded by $O(n)$.

Sensitivity Analysis. In the context of differential privacy, the sensitivity of a function is defined as the maximum difference in function output when one single element in the function domain is modified. The domain of dK-2 is a graph G. Neighbor graphs of G are all the graphs G' which differ from G by at most a single edge. Changing a single edge in G will result in one or more entries changing in the corresponding dK-2-series. Thus, the sensitivity of dK-2 is computed as the maximum number of changes in the dK-2-series among all of G's neighbor graphs.

LEMMA 1. *The sensitivity of dK-2 on a graph G, S_{dK-2}, is upper bounded by $4 \cdot d_{max} + 1$.*

PROOF. Let e be a new edge added to a graph $G = (V, E)$ between any two nodes $u, v \in V$. Once the edge e is added to G the degrees of u and v increase from d to $(d + 1)$ and from d' to $(d' + 1)$ respectively. This graph transformation produces the following changes in the dK-2 on G: the frequency k of tuple $\{d + 1, d' + 1; k\}$ gets incremented by 1 because of the new edge (u, v). For example, a new edge between A and C in Figure 1 produces an increment of the frequency k of the tuple $\{2, 3; k\}$ from $k = 1$ to $k = 2$. Furthermore, a total of $d + d'$ already present tuples need to be updated with the new degree of u and v, and so the tuples with the old degrees get decremented by a total of $d + d'$ and the tuples reflecting the new degree get incremented for a total of $d + d'$. To summarize, the overall number of changes in the dK-2 -series is $2(d + d') + 1$. In the worst case, when u and v are nodes of maximum degree d_{max}, the total number of changes in the original dK-2-series by adding an edge between u and v is upper bounded by $4 \cdot d_{max} + 1$. □

Lemma 1 derives only the upper bound of the sensitivity because, as in Definition 3 [14], it is the sufficient condition to derive the necessary amount of noise to achieve a given ϵ-privacy level. Lemma 1 shows that the sensitivity of dK-2 is high, since d_{max} has been shown to be $O(\sqrt{n})$ in measured graphs [29,43]. Note that prior work on differential privacy [9,10,15,23] generally involved functions with a much lower sensitivity, *i.e.* 1. In these cases, the low sensitivity means that the amount of noise required to generate differentially private results is very small. In contrast, the sensitivity of our function indicates that the amount of noise needed to guarantee ϵ-differential privacy in dK-2 will be high. Therefore, the accuracy of synthetic graphs generated using this method will be low. Note that if we use a higher order dK-series, *i.e.* $K \geq 3$, we would have found an even higher sensitivity value, which may further degrade the accuracy of the resulting synthetic graphs.

3.2 The dK-Perturbation Algorithm

We now introduce the dK-perturbation algorithm (dK-PA) that computes the noise to be injected into dK-2 to obtain ϵ-differential privacy [14]. In dK-PA, each element of the dK-2-series is altered based on a stochastic variable drawn from the Laplace distribution, $Lap(\lambda)$. This distribution has density function proportional to $e^{-\frac{|x|}{\lambda}}$, with mean 0 and variance $2\lambda^2$. The following theorem

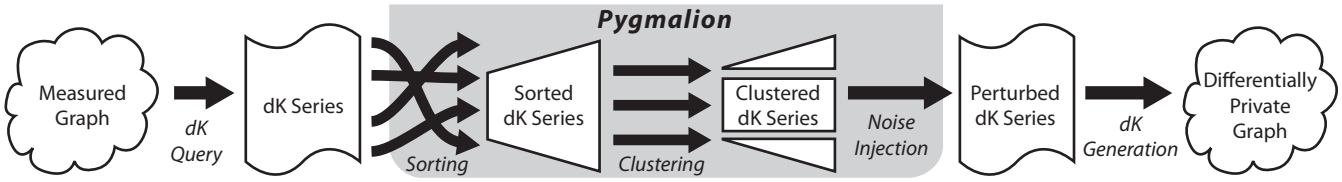


Figure 2: Overview of Pygmalion. ϵ-differential privacy is added to measured graphs after sorting and clustering the dK-2-series.

proves the conditions under which ϵ-differential privacy is guaranteed [17].

THEOREM 1. *Let $\widetilde{DK}$ be the privacy mechanism performed on dK such that $\widetilde{DK(G)} = dK(G) + Lap(\frac{S_{dK-2}}{\epsilon})^m$. For any G and G' differing by at most one edge, $\widetilde{DK}$ provides ϵ-differential privacy if:*

$$\Big| \ln \frac{Pr[\widetilde{DK(G)} = s]}{Pr[\widetilde{DK(G')} = s]} \Big| \leq \epsilon$$

PROOF. Let $s =< s_1, s_2, ..., s_m >$ be a possible output of $\widetilde{DK(G)}$ and m the number of its entries, and let G' be the graph with at most one different edge from G. Using the conditional probabilities, we have:

$$\frac{Pr[\widetilde{DK(G)} = s]}{Pr[\widetilde{DK(G')} = s]} = \prod_{i=1}^{m} \frac{Pr[\widetilde{DK(G)}_i = s_i | s_1, ...s_{i-1}]}{Pr[\widetilde{DK(G')}_i = s_i | s_1, ...s_{i-1}]},$$

since each item of the product has the first $i-1$ values of dK-2 fixed. Each s_i is the result of applying Laplacian noise calibrated by S_{dK-2}. Note that Lemma 1 has studied the sensitivity of dK-2, S_{dK-2}, under the condition that two graphs differ by at most one edge. Thus, the conditional probability is Laplacian, allowing us to derive the following inequalities:

$$\prod_{i=1}^{m} \frac{Pr[\widetilde{DK(G)}_i = s_i | s_1, ...s_{i-1}]}{Pr[\widetilde{DK(G')}_i = s_i | s_1, ...s_{i-1}]} \leq \prod_{i=1}^{m} e^{\frac{|\widetilde{DK(G)}_i - \widetilde{DK(G')}_i|}{\sigma}}$$

where σ is the scale parameter of the Laplace distribution that is $\frac{4d_{max}+1}{\epsilon}$. Thus,

$$\prod_{i=1}^{m} e^{\frac{|\widetilde{DK(G)}_i - \widetilde{DK(G')}_i|}{\sigma}} = e^{\frac{||\widetilde{DK(G)} - \widetilde{DK(G')}||_1}{\sigma}}$$

where, by definition $\widetilde{DK(G)} = dK(G) + Lap(\frac{S_{dK-2}}{\epsilon})$, and $||DK(G) - DK(G')||_1 \leq S_{dK-2}$ with $S_{dK-2} \leq 4d_{max} + 1$ as proved in Lemma 1. Thus, we have:

$$e^{\frac{||\widetilde{DK(G)} - \widetilde{DK(G')}||_1}{\sigma}} =$$

$$= e^{\frac{||dK(G) + Lap(\frac{S_{dK-2}}{\epsilon}) - dK(G') - Lap(\frac{S_{dK-2}}{\epsilon})||_1}{\sigma}} \leq e^{\frac{4d_{max}+1}{\frac{4d_{max}+1}{\epsilon}}} = e^{\epsilon}$$

and so, by applying the logarithmic function, we have that

$$\Big| \ln \frac{Pr[\widetilde{DK(G)} = s]}{Pr[\widetilde{DK(G')} = s]} \Big| \leq \epsilon$$

which concludes the proof. □

Type	Graph	Nodes	Edges
Internet	WWW	325,729	1,090,108
	AS	16,573	40,927
Facebook	Monterey Bay	14,260	93,291
	Russia	97,134	289,324
	Mexico	598,140	4,552,493
	LA	603,834	7,676,486

Table 1: Different measurement graphs used for experimental evaluation.

Theorem 1 shows that by adding noise to the dK-2-series using independent Laplace random variables calibrated by S_{dK-2} from Lemma 1, we achieve the desired ϵ-privacy.

Quantifying Accuracy. We apply the *error analysis* proposed by [25] on dK-PA to quantify the accuracy of the synthetic graphs it produces, compared to the original graphs.

DEFINITION 2. *For a perturbed dK-2-series that is generated by the privacy mechanism $\widetilde{DK}$ on a graph G, as defined in Theorem 1, the estimated error on $\widetilde{DK}$ can be computed as the expected randomization in generating $\widetilde{DK}$.*

We now quantify the expected randomization in $\widetilde{DK}$:

$$\sum_{i=1}^{m} E[(\widetilde{DK(G)}_i - dK(G)_i)^2] = mE[Lap(\frac{S_{dK-2}}{\epsilon})^2]$$

Using Lemma 1 and that $m = O(d_{max}^2)$we have:

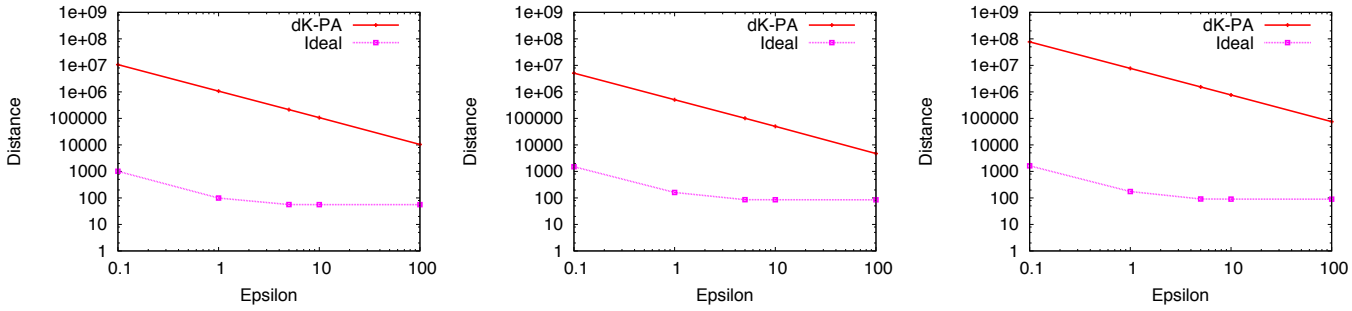
$$mE[Lap(\frac{S_{dK-2}}{\epsilon})^2] = mVar(Lap(\frac{d_{max}}{\epsilon})) = \frac{2m \cdot d_{max}^2}{\epsilon^2} = O(\frac{d_{max}^4}{\epsilon^2}).$$

This asymptotical bound shows that the noise injected by dK-PA into dK-2 scales with the fourth-degree polynomial of d_{max}. This result implies that synthetic graphs generated by dK-PA will have relatively low accuracy because of the large error introduced by the perturbation process. Furthermore, it implies that even for relatively weak privacy guarantees, dK-PA will introduce large errors that may significantly change the structure of the resulting synthetic graphs from the original.

3.3 Validation on Real Graphs

At this point, we have demonstrated analytically that the impact of adding noise to the dK-2-series using dK-PA will result in synthetic graphs that deviate significantly from the originals. In this section, we empirically evaluate the impact of adding noise to the dK-2-series by executing dK-PA on real graphs.

Methodology. To illustrate that our system is applicable to different types of graphs, we select a group of graphs that include social graphs from Facebook [41, 43], a WWW graph [3] and an AS topology graph [38] crawled on Jan 1st, 2004, which have



(a) AS (b) Russia (c) WWW

Figure 3: The noise required for different privacy levels quantified as the Euclidean distance between a graph's original and perturbed dK-2 series.

been used in prior graph mining studies [28]. The social graphs were gathered using a snowball crawl of the Facebook regional networks [43], and show graph metrics highly consistent with Facebook graphs generated using unbiased sampling techniques [21]. Table 1 lists the graphs used in our evaluation, which range from 14K nodes to 650K nodes.

We extract the dK-2-series for each graph, introduce noise using the dK-PA strategy, then compute the Euclidean distance between the perturbed dK-2-series and the original as a measure of the level of graph structural error introduced. We computed results for all graphs in Table 1, and they are consistent. For brevity, we limit ourselves to report results only for the AS graph, the WWW graph, and the Russia Facebook graph. We choose Russia to represent our social graphs because its results are representative of the other graphs, and its size does not result in extremely long run time for our experiments.

Results. Figure 3 shows that the dK-PA strategy produces a large error for small values of ϵ (*i.e.* strong privacy guarantees). We compute the error as the Euclidean distance between the original dK-2-series and the perturbed dK-2-series with dK-PA strategy. As we mentioned, the low level of accuracy is due to the large noise dK-PA injects into dK-2, resulting in a perturbed dK-2 that is significantly different from the original. The bright side is that the dK-PA strategy is robust across different datasets, and the error decreases exponentially as ϵ grows, which is shown by the linear correlation in the log-log scale plot of Figure 3.

The high error is largely due to the high sensitivity of our function dK-2. To understand the potential lower-bound on the error, we imagine a scenario where if we had a function with sensitivity of 1, then we could achieve much lower error, plotted in Figure 3 as the *Ideal* line. Note that this line is a hypothetical lower bound that is only meant to demonstrate the impact of the dK function's sensitivity on the final result. Indeed, Figure 3 shows that the loss in accuracy of our model can largely be attributed to the sensitivity of the dK-2 series.

4. PRIVACY VIA PARTITIONING

The results in the previous section demonstrate the loss of accuracy in the perturbed dK-2-series after adding noise to guarantee ϵ-differential privacy. In this section we propose a novel algorithm called Divide Randomize and Conquer (DRC) that enables more granular control over the noise injected into the dK-2-series. This qualifies DRC to support ϵ-differential privacy while also allowing for more accurate results. First, we discuss the design of DRC and prove that it does guarantee ϵ-differential privacy. Next, we investigate the amount of error introduced with this approach, and show that DRC requires significantly less noise than dK-PA to achieve an equal level of privacy. Finally, we propose an optimized version of DRC, called LDRC, and empirically verify the improved accuracy of our algorithms using measured graphs.

4.1 Divide Randomize and Conquer Algorithm

Our goal is to develop an improved privacy mechanism that significantly reduces the amount of noise that must be added to achieve a given level of ϵ-privacy. While we cannot change the fact that the sensitivity of dK-2 scales with d_{max}, our insight is to partition data in the dK-2-series into a set of small sub-series, then apply the perturbation independently to achieve ϵ-privacy within each sub-series.

If we carefully perform the partitioning to group together tuples with similar degree, we effectively reduce the value of d_{max} for each of the vast majority of sub-series. This means we can achieve ϵ-privacy on each sub-series for a fraction of the noise required to achieve ϵ-privacy across the entire series. We will then prove that ϵ-differential privacy holds across the entire dK-2-series if it holds for each of the partitioned sub-series. Thus, we produce an alternative algorithm that achieves the same level of privacy as dK-PA, while introducing significantly less noise.

We instantiate our ideas as the Divide Randomize and Conquer algorithm (DRC). The core steps of DRC are:

1. Partition (*Divide*) the dK-2-series into sub-series with specific properties;
2. Inject noise into each sub-series (*Randomize*);
3. *Conquer* the perturbed sub-series into a single dK-2-series.

In the remainder of this section we discuss the partitioning step of DRC. We first define an ordering function on dK-2 to sort tuples with similar sensitivity. The ordered dK-2 is then partitioned into contiguous and mutually disjoint sub-series. We prove that the properties of these sub-series lead to the definition of a novel sensitivity function and consequently to a novel methodology to add noise. Noise injection, conquering, and the resulting error analysis are discussed in Section 4.2.

∂ ordering on dK-2. The dK-2-series is sorted by grouping dK-tuples with numerically close pairs of degrees. In particular, the dK-tuples are sorted in the new dK-2 series, named β-series, by iteratively selecting from the original series all the tuples $\{d_x, d_y; k\}$ with degrees $(d_x \,\&\, d_y) \leq i, \forall\, i \in [1, d_{max}]$. Thus, the β-series is simply the sorted list of dK-tuples that adhere to the above inequality ordering. For example, the tuple $\{1, 2; k\}$ is

closer to $\{5, 5; k'\}$ than to $\{1, 8; k''\}$. We can formally describe this transformation with the following function:

DEFINITION 3. *Let ∂ be the sorting function on dK-2 which is formally expressed as:*

$$\partial(i) = \min_{d_x, d_y \in dK} \{\max(d_x, d_y) \geq \max(d_{x'}, d_{y'}) = \partial(i-1)\}$$

Note that $\{d_x, d_y; k\} \neq$ the first $i-1$ tuples. Thus, the ∂ function is a transformation of dK-2 such that $\partial : \Im \rightarrow \beta$ where β identifies the ordered dK-2.

Partitioning the β-Series. The β-series is partitioned into $\widetilde{m}$ sub-series, with the ith named β_i for $i \in [1, \widetilde{m}]$. The partition of β is based on two properties. First, the ∂ ordering has to be obeyed and thus each partition can only acquire *contiguous* tuples in the β-series. Second, each tuple can appear in *one and only one* sub-series. Given the ∂ ordering and the above two rules we can guarantee *mutually disjoint* and *contiguous* sub-series β_i. These two constraints are fundamental to satisfying the sensitivity properties we prove in the following Lemma 2 and Lemma 3.

Sensitivity of β_i sub-series. The sensitivity of each β_i-series can be studied following the same logic used to find the sensitivity of dK-2, by quantifying the maximum number of changes that may occur in the β_i-series due to an edge change in the graph G. Due to the ∂ ordering imposed in each sub-series, we can show that the maximum degree in each β_i plays a fundamental role in bounding its sensitivity.

LEMMA 2. *The sensitivity S_{β_i} of a sub-series β_i with tuple degrees almost equal to $d_k + 1$ is upper bounded by $4 \cdot d_k + 1$.*

The proof of this lemma is sketched because it follows the logic of Lemma 1. Due to the proposed ∂ ordering, each sub-series i is composed only of tuples where both degrees are less than or equal to a particular integer d. The worst-case (*i.e.* the maximum number of changes to the tuples in the same β_i) occurs when the tuple with degrees $d-1$ are in the same sub-series. Therefore, the maximum number of changes occur when a new edge is added between two nodes (u, v) both with degree $d-1$, after which both nodes u and v have degree d. Adding a new edge between u and v causes $d_k = d - 1$ entries in β_i to become invalid. Each invalid entry is replaced with new entry of degree d. Thus, the upper bound on the total number of changes is $2 \cdot d_k$ deletions, $2 \cdot d_k$ additions, and one new edge, with the total being $4 \cdot d_k + 1$.

Given the partitioning approach and the imposed ∂ ordering across sub-series, we are able to exploit further properties on the β_is-series. In particular, the sensitivity of any β_i is independent from the location where the change occurs in the graph. Conversely, the sensitivity of a particular partition is dependent on the tuple with the highest degree values, as proved in Lemma 2. Therefore:

LEMMA 3. *The sensitivity of any β_i is independent by the sensitivity of any other β_j with $i \neq j$.*

PROOF. The proof proceeds by contradiction from the following assumption: *the sensitivity of a β_i is impacted by a change occurring in a β_j with $i \neq j$.* Without loss of generality, assume $i < j$, and $\partial(i')$ is a tuple in β_i and $\partial(j')$ is a tuple in β_j, as from Definition 3. Assume that an edge is formed between a node x with corresponding tuples $< \partial(i'), \partial(i'+1), .. > \in \beta_i$ and a node y with corresponding tuples $< \partial(j'), \partial(j'+1)... > \in \beta_j$. The maximum number of changes that can occur due to this event is bounded by the degree values of x and y. Let d be the new degree of x. The maximum number of tuples that can change in β_i are $d-1$ tuples that get deleted and d that get added, which is $< 2 \cdot d$. Symmetrically, let b be the new degree of y so the maximum number of tuples that can change in β_j is $< 2 \cdot b$. Even if d and b are equal to the maximum degree value d_k within their sub-series, as demanded in Lemma 2, the number of changes involved in each sub-series is $2 \cdot d_k < 4 \cdot d_k + 1$ which means that the sensitivity of both β_i and β_j are not mutually effected, which contradicts the hypothesis. □

4.2 Theoretical Analysis

This section is devoted to the theoretical analysis of the privacy and accuracy properties the DRC approach achieves. First, we prove that ϵ-differential privacy can be applied to each sub-series created during the partitioning phase of DRC. Next, we build on this result to prove that the individual differentially private sub-series' can be reunified into a complete dK-2-series that is also ϵ-differentially private. Lastly, we perform error analysis on DRC and compare the results to dK-PA.

Analyzing ϵ-Privacy in β_is. We now quantify the privacy of each β_i and prove that they satisfy ϵ-differential privacy.

THEOREM 2. *For each cluster β_i with $i = 1, .., \widetilde{m}$, let $\widehat{\beta}_i$ be a novel privacy mechanism on β_i such that $\widehat{\beta}_i = \beta_i + Lap(\frac{S_{\beta_i}}{\epsilon})^{|\beta_i|}$. Then, for all sub-series β_i and β_i' derived from graphs G and G' that differ by at most one edge, $\widehat{\beta}_i$ satisfies ϵ-differential privacy if:*

$$\Big| \ln \frac{Pr[\widehat{\beta}_i = s]}{Pr[\widehat{\beta}_i' = s]} \Big| \leq \epsilon$$

PROOF. Let m^* be the the cardinality of cluster β_i. Let G' be a graph with at most one edge different from G. Let s_j be the j^{th} item of the $\widehat{\beta}_i$-series, that is $\widehat{\beta}_i[j] = s_j$. Using the conditional probability on s_j we can write:

$$\frac{Pr[\widehat{\beta}_i = s]}{Pr[\widehat{\beta}_i' = s]} = \prod_{j=1}^{m^*} \frac{Pr[\widehat{\beta}_i[j] = s_j | s_1, ...s_{j-1}]}{Pr[\widehat{\beta}_i'[j] = s_j | s_1, ...s_{j-1}]}$$

Each item of the product has the first $j-1$ tuples of the $\widehat{\beta}_i$-series fixed. Each s_j is the result of the Laplace noise that has been calibrated for β_i based on its sensitivity, as calculated using in Lemma 2. The sensitivity of this function is derived under the assumption that the two graphs have, at most, one edge difference. Thus, the conditional probabilities are Laplacians, which allows us to derive the following inequalities:

$$\prod_{j=1}^{m^*} \frac{Pr[\widehat{\beta}_i[j] = s_j | s_1, ...s_{j-1}]}{Pr[\widehat{\beta}_i'[j] = s_j | s_1, ...s_{j-1}]} \leq \prod_{j=1}^{m^*} e^{\frac{|\widehat{\beta}_i[j] - \widehat{\beta}_i'[j]|}{\sigma}}$$

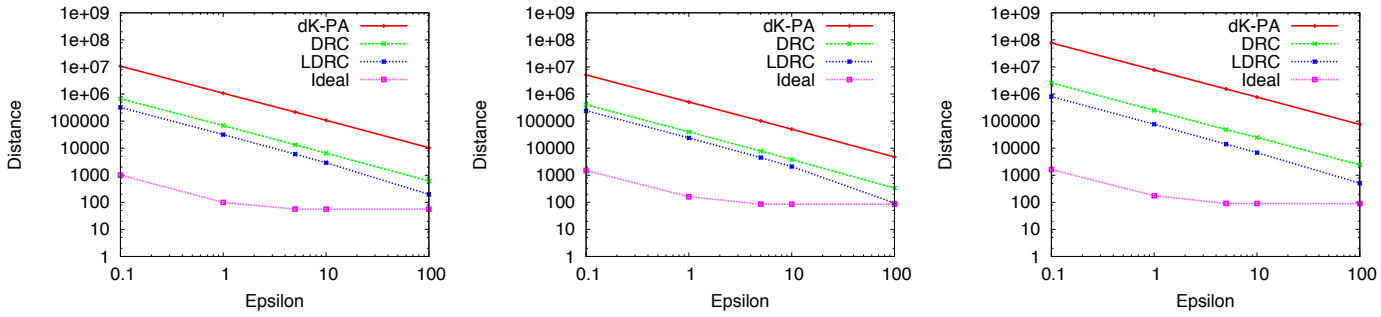
By definition $\widehat{\beta}_i = \beta_i + Lap(\frac{S_{\beta_i}}{\epsilon})^{|\beta_i|}$ and by Lemma 2 $||\beta_i - \beta_i'||_1 \leq S_{\beta_i}$ with $S_{\beta_i} \leq 4d_{k_i} + 1$. Let σ_i be the scale parameter of the Laplacian noise applied in each cluster i, thus:

$$\prod_{j=1}^{m^*} e^{\frac{|\widehat{\beta}_i[j] - \widehat{\beta}_i'[j]|}{\sigma}} = e^{\frac{||\widehat{\beta}_i - \widehat{\beta}_i'||_1}{\sigma}}$$

$$= e^{\frac{||\widehat{\beta}_i + Lap(\frac{S_{\beta_i}}{\epsilon}) - \widehat{\beta}_i' - Lap(\frac{S_{\beta_i}}{\epsilon})||_1}{\sigma}} = e^{\frac{||\beta_i - \beta_i'||_1}{\sigma}} \leq e^{\frac{\frac{4d_{m_i}+1}{4d_{m_i}+1}}{\epsilon}}$$

Finally, by applying the logarithmic function the theorem statement is proved. □

Theorem 2 shows that adding noise does achieve provable ϵ-differential privacy on each cluster. In particular, we prove that by



(a) AS (b) Russia (c) WWW

Figure 4: Euclidean distances of the dK-2-series of different ϵ-Differential Privacy strategies on three real graphs.

only leveraging m^* independent Laplace random variables, with parameter $\lambda = (\frac{S_{\beta_i}}{\epsilon})$, it is possible to generate sufficient noise per cluster to satisfy the privacy requirement.

Conquering ϵ-privacy into $\cup_i \widehat{\beta}_i$. Our next task is to leverage the proved ϵ-differential privacy of each independent $\widehat{\beta}_i$ to guarantee privacy on the entire perturbed $\widehat{\beta}$-series$= \cup_i \widehat{\beta}_i$. In order to achieve this goal a further step is required, shown in the following corollary.

COROLLARY 1. *The amount of information an attacker can learn on $\widehat{\beta}_i$ by observing any $\widehat{\beta}_j$ with $i \neq j$ is null.*

This proof considers only two sub-series for simplicity. Given Lemma 3, this proof can be extended to any number of clusters.

PROOF. Let A and B be two sub-series built out of our partition strategy and let $\widehat{A}$ and $\widehat{B}$ be their ϵ-differentially private projection as proved in Theorem 2. Finally, let a and b be events on $\widehat{A}$ and $\widehat{B}$, respectively. Through the Shannon Entropy Theory we quantify the information a sub-series could exploit on another sub-series. In particular, the Mutual Information

$$I(\widehat{A}; \widehat{B}) = \sum_{a,b} p(a,b) \log \frac{p(a,b)}{p(a)p(b)}$$

is the amount of information an attacker can infer on $\widehat{A}$ by observing $\widehat{B}$. By construction the sensitivity of the sub-series A is independent from the sensitivity of the sub-series B, as proved in Lemma 3. This means that the sub-series A is perturbed by a Laplace random process with parameter λ_A that is independent from the Laplace random process acting on B, as consequence of Lemma 2. Thus, this independence property directly implies that the Mutual Information $I(\widehat{A}, \widehat{B}) = 0$, that is, an attacker gains no information on $\widehat{A}$ by observing $\widehat{B}$, which concludes the proof. □

The properties derived on the different β_is are sufficient to begin the **conquer phase** of our DRC approach. The goal of the conquer phase is to unify the $\widehat{\beta}_i$s such that the union set inherits the ϵ-privacy guarantees from the individual sub-series.

THEOREM 3. *Given $\widetilde{m}$ different sub-series $\widehat{\beta}_i$ with $i = 1, ..., \widetilde{m}$, the result of the DRC conquer strategy $\cup_i \beta_i$ satisfies the ϵ-differential privacy property.*

PROOF. The DRC strategy produces $\widetilde{m}$ ϵ-differentially private sub-series $\widehat{\beta}_i$, as proved in Theorem 2. Each β_i satisfies Lemma 2 and Lemma 3, and any combination of $\widehat{\beta}_i$s satisfies Corollary 1. The privacy independence property, from Corollary 1, implies that $\cup_i \widehat{\beta}_i$ satisfies the ϵ-Differential Privacy property. □

Thus, we have proven that our perturbed dK-2, $\cup_i \widehat{\beta}_i$, satisfies the ϵ-differential privacy requirement. DRC achieves a tighter bound on noise than dk-PA due to the properties from Lemmas 2 and 3.

Error Analysis. We now quantify the error introduced to dK-2 via our DRC strategy. Error analysis on DRC is complicated because our algorithm does not specify the number of clusters to generate during partitioning. Instead, our clustering approach is general, and covers any possible set of cuts on the β-series such that the resulting sub-series differ in cardinality and sensitivity from each other, so long as they respect Lemmas 2 and 3. Therefore, in order to provide an error analysis that covers any possible clustering of the β-series we have to study both the lower and the upper bound of the error injected into those series.

DEFINITION 4. *The error estimation of the union of the $\widehat{\beta}_i$s under the ∂ ordering on dK-2 of a graph G can be computed as the expected randomization in generating $\widehat{\beta} = \cup_i \widehat{\beta}_i$.*

The expected randomization in $\widehat{\beta}$ is quantified as

$$\sum_{i=1}^{\widetilde{m}} E\left(\sum_j (\widehat{\beta}_i[j] - \beta_i[j])^2\right) = \sum_{i=1}^{\widetilde{m}} |\beta_i| E[Lap(\frac{S_{\beta_i}}{\epsilon})^2]$$

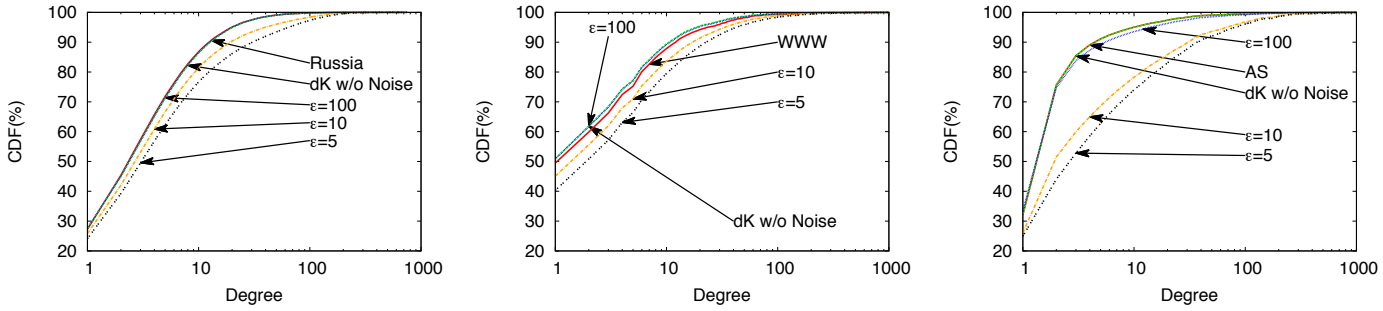
The lower bound is found when each S_{β_i} have the same minimum value, which is 1, and thus

$$\sum_{i=1}^{\widetilde{m}} |\beta_i| E[Lap(\frac{S_{\beta_i}}{\epsilon})^2] \geq d_{max}^2 Var(Lap(\frac{1}{\epsilon})) = \Omega(\frac{d_{max}^2}{\epsilon^2})$$

Note that the considered minimum, *i.e.* 1, happens only when a graph of nodes with zero degree is considered, and after adding an edge S_β is 1. The upper bound is found when each S_{β_i} have the maximum value that, as proved in Lemma 2, is $O(d_{max})$, and thus

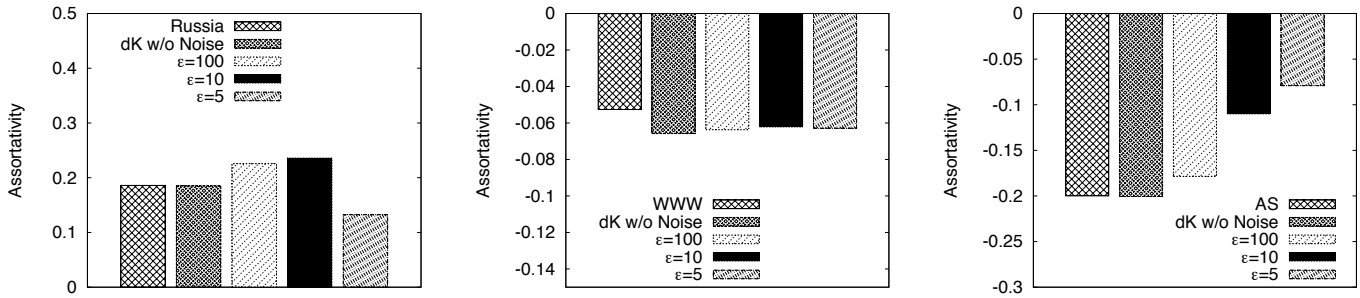
$$\sum_{i=1}^{\widetilde{m}} |\beta_i| E[Lap(\frac{S_{\beta_i}}{\epsilon})^2] \leq d_{max}^2 Var(Lap(\frac{d_{max}}{\epsilon})) = O(\frac{d_{max}^4}{\epsilon^2})$$

The worst-case error level of DRC is equal to that of dK-PA. However, depending on graph structure, the error level can decrease down to $\Omega(\frac{d_{max}^2}{\epsilon^2})$. As we demonstrate in the next section, real graphs exhibit error rates towards the lower bound. Thus, in practice, DRC performs much better than dK-PA.



(a) Russia (b) WWW (c) AS

Figure 5: Degree distribution of three real measured graphs, *i.e.* Russia, WWW and AS, each compared to the dK-synthetic graph without noise and Pygmalion synthetic graphs with different ϵ values.



(a) Russia (b) WWW (c) AS

Figure 6: Assortativity of three real measured graphs, *i.e.* Russia, WWW and AS, each compared to the dK-synthetic graph without noise and Pygmalion synthetic graphs with different ϵ values.

4.3 Evaluating and Optimizing DRC

To quantify the improvement DRC achieves over the dK-PA strategy, we compare the results of applying each algorithm on our graphs. As before in Section 3.3, we quantify error using the Euclidean distances between each of their dK-2-series and the dK-2-series of the original graph. As seen in Figure 4, DRC reduces the Euclidean distance by one order of magnitude for different graphs and a range of ϵ values. As is the case for dK-PA, error introduced by DRC decreases exponentially as the value of ϵ increases, which is clear from the linear correlation in the log-log scale plot of Figure 4.

Further Optimization with LDRC. Despite its improvement over dK-PA, DRC is still quite far from the idealized function in terms of error (see Figure 4). We apply a prior result from [25] that proves how to use isotonic regression [6], *i.e.* evenly "smooth" out the introduced noise across tuples, without breaking differential privacy properties. This technique enables a reduction of the error introduced in the dK-2-series by another constant factor.

Formally, given a vector p of length p^*, the goal is to determine a new vector p' of the same length which minimizes the L_2 norm, *i.e.* $||p - p'||_2$. The minimization problem has the following constraints: $p'[i] \leq p'[i+1]$ for $1 \leq i < p^*$. Let $p[i, j]$ be a sub-vector of length $j - i + 1$, that is: $< p[i], ..., p[j] >$. Let define $M[i, j]$ as the mean of this sub-vector, *i.e.* $M[i, j] = \sum_{k=i}^{j} p[k]/(j - i + 1)$.

THEOREM 4. *[6] The minimum L_2 vector, p', is unique and is equal to $p'[k] = \widetilde{M_k}$, with:*

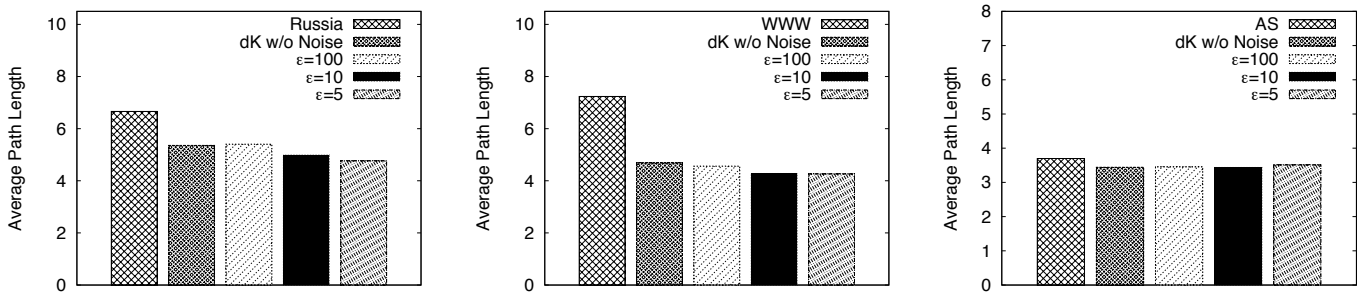
$$\widetilde{M_k} = min_{j \in [k,p^*]} max_{i \in [1,j]} M[i, j]$$

We apply this technique on the set of all tuples produced by DRC. We refer to it as the L_2 minimization Divide Randomize and Conquer algorithm, or LDRC. We include LDRC in our comparison of algorithms in Figure 4, and see that LDRC provides roughly another 50% reduction in error over the DRC algorithm. Since it consistently outperforms our other algorithms, we use LDRC as the algorithm inside the Pygmalion graph model.

Implications. Finally, we note that our DRC partition technique is general, and has potential implications in other contexts where it is desirable to achieve differential privacy with lower levels of injected noise. More specifically, it can serve to reduce the amount of perturbation necessary when the required perturbation is a function of a parameter that varies significantly across values in the dataset.

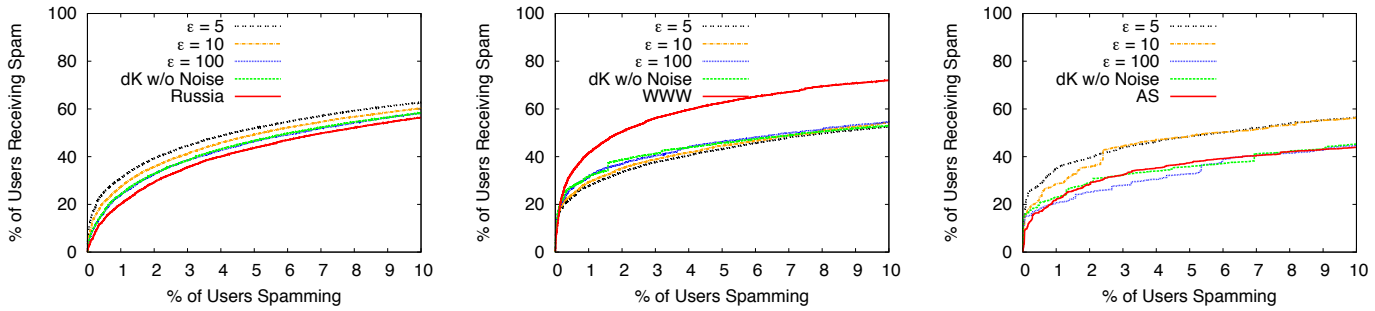
5. END-TO-END GRAPH SIMILARITY

We have already quantified the level of similarity between real and synthetic graphs by computing the Euclidean distances between their respective dK-series datasets. These values represent the distortion in the statistical representation of a graph, *i.e.* the dK-series, but do not capture the ultimate impact of the added noise on graph structure. In this section, we evaluate how well Pygmalion preserves a graph's structural properties by comparing Pygmalion's differentially private synthetic graphs against the originals in terms of both graph metrics and outcomes in application-level tests. Strong structural similarity in these results would establish the feasibility of using these differentially private synthetic graphs in real research analysis and experiments.



(a) Russia (b) WWW (c) AS

Figure 7: Average path length of three real measured graphs, *i.e.* Russia, WWW and AS, each compared to the dK-synthetic graph without noise and Pygmalion synthetic graphs with different ϵ values.



(a) Russia (b) WWW (c) AS

Figure 8: Reliable Email (RE) experiment run on three real measured graphs, *i.e.* Russia, WWW and AS, each compared with the dK-synthetic graph without noise and Pygmalion synthetic graphs with different ϵ values.

5.1 Graph Metrics

Our evaluation includes two classes of graph metrics. One group includes *degree-based metrics* such as: Average Node Degree, Degree Distribution, Joint Degree Distribution and Assortativity. These are basic topological metrics that characterize how degrees are distributed among nodes and how nodes with particular degree connect with each other. The second group includes *node separation metrics* that quantify the interconnectivity and density of the overall graph. This group includes metrics such as Graph Diameter, Radius and Average Path Length.

For our evaluation purposes, we always use our most advanced algorithm, *i.e.* Pygmalion LDRC. We only focus on Pygmalion LDRC, because there are practical problems in generating large graphs from dK values after significant noise has been added. As shown earlier, the dK-PA model introduces the highest noise. In fact, errors introduced by dK-PA are so large that the generator fails when trying to generate large graphs with the resulting noisy dK distributions.

We generate ϵ-private graphs for $\epsilon \in [5, 100]$, and compare the graph metrics of the resulting synthetic graphs against those of the original graph, and a synthetic graph generated by the dK model with no additional noise added. We limit ourselves to ϵ-private graphs with $\epsilon \in [5, 100]$ because of two reasons. First, we aim to find the ϵ value that contributes to a smallest noise such that it is statistically similar to the synthetic dK-2 graph with no privacy enforced. This way, we can indirectly quantify the level of privacy introduced by a pure synthetic graph with no additional steps taken to improve privacy. This by itself is a potentially interesting result. In particular, we obtain this property only when ϵ is equal to 100. Second, the dK-2 distribution is a very sensitive function and it naturally requires a high level of noise to provide strong levels of privacy guarantees. Unfortunately, very small values of ϵ require larger noise values, thus producing synthetic graphs that are extremely different in structure from the original. Finally, for $\epsilon < 1$, the required noise level is so high for larger graphs, that the dK graph generator fails to produce synthetic graphs that match the resulting dK distributions. This is clearly a limitation of the current system, one that we hope will be removed with the discovery of less sensitive models and optimization techniques to further reduce noise required for ϵ-differential privacy.

As we mentioned, our results are highly consistent across our pool of graphs (Table 1), and we only report experimental results on three graphs: the Russia Facebook graph, the AS graph and the WWW graph.

Degree-based Metrics. These metrics are fundamental in understanding the statistical properties of node degrees and how nodes connect to each other to form specific topological structures. Out of the four metrics mentioned above, we report results for Degree-Distribution (which supersedes average node degree) and Assortativity (which is related to joint degree distribution).

Degree Distributions. Figure 5 compares the node degree CDFs. For each of the Russia, WWW, and AS graphs, the degree distributions of both the Pygmalion (ϵ=100) graph and the dK-synthetic graph very closely match the degree distribution of the original graphs. When we increase the strength of the privacy guarantees, *i.e.* smaller ϵ values of 5 and 10, the accuracy of the synthetic de-

gree distribution progressively decreases. For example, both the Russia and WWW graphs show a small deviation from the original distribution even for $\epsilon = 5$. Across all models for these two graphs, the worst-case degree distribution deviation is still within 10% of the original.

The AS graph, on the other hand, shows a slightly different behavior. For small ϵ values, *i.e.* $\epsilon = 5$ and $\epsilon = 10$, the largest error is within 35% from the original graph values. The AS graph shows a different behavior because a small number of high degree nodes connect the majority of other nodes. Thus, when the privacy perturbation hits those high-degree nodes, it can produce structural changes that send ripples through the rest of the graph.

Assortativity. Figure 6 reports the results of the assortative metric computed on both real and synthetic graphs for each of the three graphs (Russia, WWW and AS). The assortativity metric describes the degree with which nodes with similar degree are connected to each other. Positive assortativity value denotes a positive correlation between the degrees of connected nodes, and negative values indicate anti-correlation. Note that both the WWW and AS graphs show negative assortativity (Figure 6(b) and Figure 6(c)).

As with the degree distribution results, for each of our graphs (Russia, WWW, and AS), assortativity results from synthetic graphs for $\epsilon = 100$ and those from the dK-series closely match results from the original graphs. As we increase the level of privacy protection, the results get slightly further from the original values. For example, using $\epsilon = 5$ on Russia produces an error less than 0.05 on the assortativity value. The same ϵ value for the WWW graph produces negligible error on assortativity. Assortativity results on the AS graph are also consistent with degree distribution results. Under high privacy requirements, *i.e.* $\epsilon = 5$, error on assortativity reaches 0.12.

Node Separation Metrics. For brevity, we report only the Average Path Length as a representative of the node separation metrics. Figure 7 shows the Average Path Length (APL) values computed on Russia, WWW and AS compared to the APL values on their synthetic graphs. On Russia and WWW, APL results denote a moderate level of error (higher when compared to results for the earlier graph metrics). We can see that the error is mainly introduced by the impreciseness of the dK-model, since the synthetic graph from the dK-series with no noise shows the same error. In comparison, the error introduced by strengthening privacy (and hence decreasing ϵ) is relatively small. This is encouraging, because we can eliminate the bulk of the error by moving from dK-2 to a more accurate model, *e.g.* dK-3.

As with previous experiments, the AS graph shows a slightly different behavior. In this case, all of our synthetic graphs do a good job of reproducing the average path length value of the AS graph.

Summary. Our experimental analysis shows that synthetic graphs generated by Pygmalion exhibit structural features that provide a good match to those of the original graphs. As expected, increasing the strength of privacy guarantees introduces more noise into the structure of the synthetic graphs, producing graph metrics with higher deviation from the original graphs. These observations are consistent across social, web, and Internet topology graphs.

Overall, these results are very encouraging. They show that we are able to effectively navigate the tradeoff between accuracy and privacy by carefully calibrating the ϵ values. The fact that significant changes in ϵ values do not dramatically change the graph structure means owners of datasets can guarantee reasonable levels of privacy protection and still distribute meaningful graphs that match the original graphs in structure.

5.2 Application Results

For a synthetic graph to be usable in research, ultimately it must produce the same results in application-level experiments as the original graph it is replacing. To quantify the end-to-end impact of trading graph similarity for privacy protection, we compare the results of running two real world applications on both differentially private synthetic graphs and the original graphs. We implement two applications that are highly dependent on graph structure: Reliable Email (RE) [20] and Influence Maximization [11].

Reliable Email. RE [20] is an email spam filter that relies on a user's social network to filter and block spam. One way to evaluate the security of RE is to compute the number of users in a network who can be spammed by a fixed number of compromised friends in the social network. This experiment depends on the structure of the network, and is a useful way to evaluate whether Pygmalion graphs can be true substitutes for measurement graphs in research experiments.

Figure 8 shows the portion of the nodes flooded with spam as we increase the number of malicious spammers, using different graphs as the underlying social network topology. We show results on the usual three graphs, Russia, WWW and AS. On the Russia Facebook graph, all synthetic graphs closely follow the original graph. Even in the case of the strongest privacy setting, *i.e.* $\epsilon = 5$, the difference between the synthetic graph result and those of the original is at most 10%. For both the WWW and AS graphs, all synthetic graphs with and without noise produce results within 20% of the original graphs.

Influence Maximization. The influence maximization problem tries to locate users in the network who can most quickly spread information through the network. This problem is most commonly associated with advertisements and public relations campaigns. Evaluating a solution to this problem includes two steps. First, the solution must identify the nodes who can maximize influence in the network. Second, it must model the spread of influence through the network to quantify how many users the influence has ultimately reached.

For our purposes, we use a recently proposed heuristic for influence maximization that minimizes computation. The heuristic is called the Degree Discount method [11], and is able to find the most influential nodes, called "seeds," on a given graph. Starting from those seed nodes, we run three different influence dissemination models: Linear threshold (LT), Independent Cascade (IC) and Weighted Cascade (WC), to determine the total number of users in the network influenced by the campaign. We use source code we obtained from the authors. However, significant memory overhead in the code meant that we had to limit our experiments to smaller graphs. Therefore, we use the MontereyBay Facebook graph and the AS network topology graph in this experiment.

For both AS and MontereyBay graphs and each of the three influence dissemination models, Figure 9 shows the expected number of influenced nodes when increasing the number of initial seed nodes. While the actual percentage of users influenced varies across dissemination models, there are clear and visible trends. Results on the AS graph in Figures 9(a), 9(b), 9(c) all show that Pygmalion with $\epsilon = 100$ and the dK-synthetic graph without noise are almost identical to the original AS graph under all three dissemination models. Graphs with stronger protection, Pygmalion $\epsilon = 10$ and $\epsilon = 5$, progressively diverge from the results of the AS graph. Results on the MontereyBay graph are shown in Figures 9(d), 9(e), 9(f), and are quite similar to those on the AS graph. They confirm that Pygmalion $\epsilon = 100$ produces near perfect results, but higher

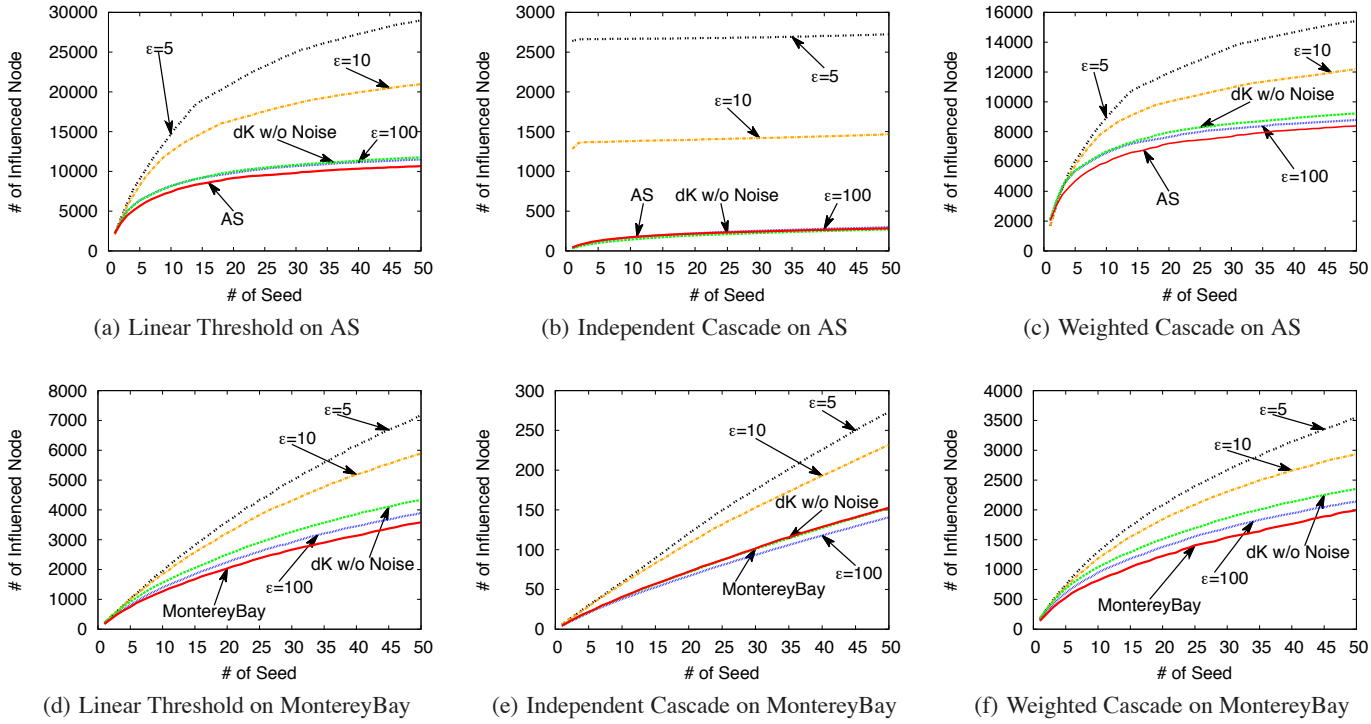


Figure 9: Results of the Degree Discount Influence Maximization algorithm on the AS and MontereyBay graphs, compared to dK graphs without added noise, and Pygmalion synthetic graphs with different ϵ values.

privacy protection increases the deviations from results on the original MontereyBay graph.

5.3 Summary

We have used both popular graph metrics and application-level tests to evaluate the feasibility of using differentially private synthetic graphs in research. Our tests are not comprehensive, and cannot capture all graph metrics or application-level experiments. However, they are instructive because they show the observable impact on graph structure and research results when we replace real graphs with differentially private Pygmalion graphs.

Our results consistently show that Pygmalion introduces limited impact as a result of adding noise to guarantee privacy. In fact, many of the largest errors can be attributed to limitations of the dK-2 series. Given the significant demand for realistic graphs in the research community, we expect that generator algorithms for more complex dK models will be discovered soon. Moving to those models, *e.g.* dK-3, will eliminate a significant source of error in these results.

6. CONCLUSION

We study the problem of developing a flexible graph privacy mechanism that preserves graph structures while providing user-specified levels of privacy guarantees. We introduce *Pygmalion*, a differentially-private graph model that aims these goals using the dK-series as a graph transformation function. First, we use analysis to show that this function has a high sensitivity, *i.e.* applied naively, it requires addition of high levels of noise to obtain privacy guarantees. We confirm this on both social and Internet graphs. Second, we develop and prove a partitioned privacy technique where differential privacy is achieved as a whole when it is achieved in each data cluster. This effectively reduces the level of noise necessary to attain a given level of privacy.

We evaluate our model on numerous graphs that range in size from 14K nodes to 1.7 million nodes. Our partitioned privacy technique reduces the required noise by an order of magnitude. For moderate to weak levels of privacy guarantees, the resulting synthetic graphs closely match the original graphs in both graph structure and behavior under application-level experiments.

We believe our results represent a promising first step towards enabling open access to realistic graphs with privacy guarantees. The accuracy of our current model is fundamentally limited by both the degree of descriptiveness of dK-2 series, and the high noise necessary to inject privacy properties. There are two ways to improve our results. One way is to use a more descriptive, higher-order dK model, under the assumption that its sensitivity is reasonable low. While generators for higher order dK-models are still unknown, our techniques are general, and can be applied to obtain more accurate models as higher-order dK generators are discovered. Another way to improve is to discover a function (or model) of graph structure with much lower sensitivity. If such a function exists, it can potentially lower the noise required for a given privacy level by orders of magnitude.

7. ACKNOWLEDGMENTS

We thank the anonymous reviewers for their comments. This material is based in part upon work supported by the National Science Foundation under grants IIS-0916307 and IIS-847925. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. This work is also supported in part by South Korea National Research Foundation under the World Class University program.

8. REFERENCES

[1] AGGARWAL, G., FEDER, T., KENTHAPADI, K., MOTWANI, R., PANIGRAHY, R., THOMAS, D., AND ZHU, A. Approximation algorithms for k-anonymity. *Journal of Privacy Technology* (2005).

[2] AHN, Y.-Y., HAN, S., KWAK, H., MOON, S., AND JEONG, H. Analysis of topological characteristics of huge online social networking services. In *Proc. of WWW* (2007).

[3] ALBERT, R., JEONG, H., AND BARABÁSI, A.-L. Internet: Diameter of the World-Wide Web. *Nature 401*, 6749 (1999), 130–131.

[4] ALBERT, R., JEONG, H., AND BARABASI, A.-L. Error and attack tolerance of complex networks. *Nature 406* (July 2000).

[5] BACKSTROM, L., DWORK, C., AND KLEINBERG, J. M. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *Proc. of WWW* (2007), pp. 181–190.

[6] BARLOW, R. E., AND BRUNK, H. D. The isotonic regression problem and its dual. *Journal of the American Statistical Association 67*, 337 (1972), 140–147.

[7] BELLARE, M., BOLDYREVA, A., AND O'NEILL, A. Deterministic and efficiently searchable encryption. In *Proc. of CRYPTO* (2007), vol. 4622, pp. 535–552.

[8] BHAGAT, S., CORMODE, G., KRISHNAMURTHY, B., AND SRIVASTAVA, D. Class-based graph anonymization for social network data. *Proc. VLDB Endow. 2* (August 2009), 766–777.

[9] BLUM, A., DWORK, C., MCSHERRY, F., AND NISSIM, K. Practical privacy: the sulq framework. In *Proc. of PODS* (2005).

[10] BLUM, A., LIGETT, K., AND ROTH, A. A learning theory approach to non-interactive database privacy. In *Proc. of STOC* (2008).

[11] CHEN, W., WANG, Y., AND YANG, S. Efficient influence maximization in social networks. In *Proc. of KDD* (2009).

[12] CHUN, H., KWAK, H., EOM, Y.-H., AHN, Y.-Y., MOON, S., AND JEONG, H. Comparison of online social relations in volume vs interaction: a case study of Cyworld. In *Proc. of IMC* (2008).

[13] DIMITROPOULOS, X., KRIOUKOV, D., VAHDAT, A., AND RILEY, G. Graph annotations in modeling complex network topologies. *ACM Trans. Model. Comput. Simul. 19* (November 2009).

[14] DWORK, C. Differential privacy. In *Proc. of ICALP* (2006).

[15] DWORK, C. Differential privacy: A survey of results. In *Theory and Applications of Models of Computation* (2008).

[16] DWORK, C., KENTHAPADI, K., MCSHERRY, F., MIRONOV, I., AND NAOR, M. Our data, ourselves: Privacy via distributed noise generation. In *Proc. of EUROCRYPT* (2006).

[17] DWORK, C., MCSHERRY, F., NISSIM, K., AND SMITH, A. Calibrating noise to sensitivity in private data analysis. In *Proc. of IACR TCC* (2006).

[18] FRIEDMAN, A., WOLFF, R., AND SCHUSTER, A. Providing k-anonymity in data mining. *The VLDB Journal* (2008).

[19] FU, Y., CHEN, Z., KORU, G., AND GANGOPADHYAY, A. A privacy protection technique for publishing data mining models and research data. *ACM Trans. Manage. Inf. Syst. 1* (December 2010), 7:1–7:20.

[20] GARRISS, S., KAMINSKY, M., FREEDMAN, M. J., KARP, B., MAZIÈRES, D., AND YU, H. Re: Reliable email. In *Proc. of NSDI* (2006).

[21] GJOKA, M., KURANT, M., BUTTS, C. T., AND MARKOPOULOU, A. Walking in Facebook: A case study of unbiased sampling of OSNs. In *Proc. of INFOCOM* (2010).

[22] GUPTA, A., LIGETT, K., MCSHERRY, F., ROTH, A., AND TALWAR, K. Differentially private combinatorial optimization. In *Proc. of SODA* (2010).

[23] HAY, M., LI, C., MIKLAU, G., AND JENSEN, D. Accurate estimation of the degree distribution of private networks. In *Proc. of IEEE ICDM* (2009).

[24] HAY, M., MIKLAU, G., JENSEN, D., TOWSLEY, D., AND WEIS, P. Resisting structural re-identification in anonymized social networks. In *Proc. of VLDB* (2008).

[25] HAY, M., RASTOGI, V., MIKLAU, G., AND SUCIU, D. Boosting the accuracy of differentially private histograms through consistency. In *Proc. of VLDB* (2010).

[26] JIANG, J., WILSON, C., WANG, X., HUANG, P., SHA, W., DAI, Y., AND ZHAO, B. Y. Understanding latent interactions in online social networks. In *Proc. of IMC* (Nov. 2010).

[27] KUMAR, R. A. Structure and evolution of online social networks. In *Proc. of KDD* (2006).

[28] KWAK, H., CHOI, Y., EOM, Y.-H., JEONG, H., AND MOON, S. Mining communities in networks: a solution for consistency and its evaluation. In *Proc. of IMC* (2009).

[29] KWAK, H., LEE, C., PARK, H., AND MOON, S. What is twitter, a social network or a news media? In *Proc. of WWW* (2010).

[30] LIU, K., AND TERZI, E. Towards identity anonymization on graphs. In *Proc. of SIGMOD* (2008).

[31] MAHADEVAN, P., KRIOUKOV, D., FALL, K., , AND VAHDAT, A. Systematic topology analysis and generation using degree correlations. In *Proc. of SIGCOMM* (2006).

[32] MCSHERRY, F., AND MAHAJAN, R. Differentially-private network trace analysis. In *Proc. of SIGCOMM* (October 2010).

[33] MEYERSON, A., AND WILLIAMS, R. On the complexity of optimal k-anonymity. In *Proc. of PODS* (2004).

[34] MIR, D., AND WRIGHT, R. A differentially private graph estimator. In *Proc. of ICDMW '09*. (December 2009), pp. 122 –129.

[35] NARAYANAN, A., AND SHMATIKOV, V. Robust de-anonymization of large sparse datasets. In *Proc. of IEEE S&P* (2008).

[36] NARAYANAN, A., AND SHMATIKOV, V. De-anonymizing social networks. In *Proc. of IEEE S&P* (2009).

[37] NISSIM, K., RASKHODNIKOVA, S., AND SMITH, A. Smooth sensitivity and sampling in private data analysis. In *Proc. of STOC* (2007).

[38] OLIVEIRA, R. V., ZHANG, B., AND ZHANG, L. Observing the evolution of internet as topology. In *Proc. of SIGCOMM* (2007).

[39] PUTTASWAMY, K. P. N., SALA, A., AND ZHAO, B. Y. Starclique: Guaranteeing user privacy in social networks against intersection attacks. In *Proc. of ACM CoNEXT* (2009).

[40] RASTOGI, V., AND NATH, S. Differentially private aggregation of distributed time-series with transformation and encryption. In *Proc. of SIGMOD* (2010).

[41] SALA, A., CAO, L., WILSON, C., ZHENG, H., AND ZHAO, B. Y. Measurement-calibrated graph models for social network experiments. In *Proc. of WWW* (2010).

[42] SWEENEY, L. k-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 10* (2002), 557–570.

[43] WILSON, C., BOE, B., SALA, A., PUTTASWAMY, K. P. N., AND ZHAO, B. Y. User interactions in social networks and their implications. In *Proc. of EuroSys* (2009).

[44] XIAO, X., WANG, G., AND GEHRKE, J. Differential privacy via wavelet transforms. *IEEE Trans. on Knowledge and Data Engineering* (2010).

[45] ZHELEVA, E., AND GETOOR, L. Preserving the privacy of sensitive relationships in graph data. In *Proc. of PinKDD* (2008), pp. 153–171.

[46] ZHOU, B., AND PEI, J. Preserving privacy in social networks against neighborhood attacks. In *Proc. of ICDE* (2008).

[47] ZIMMER, M. Facebook data of 1.2 million users from 2005 released: Limited exposure, but very problematic. Blog, February 2011. `http://michaelzimmer.org`.

[48] ZOU, L., CHEN, L., AND OZSU, M. T. K-automorphism: A general framework for privacy preserving network publication. In *Proc. of VLDB* (2009).

Summary Review Documentation for

"Sharing Graphs using Differentially Private Graph Models"

Authors: A. Sala, X. Zhao, C. Wilson, H. Zheng, B. Zhao

Reviewer #1

Strengths: Focuses on the important and timely problem of privacy-preserving graph anonymization. This is a very general and interesting problem. The paper proposes a novel approach to apply differential-privacy concepts on graphs. It produces anonymized graphs with specific privacy guarantees. The proposed method is thoroughly analyzed and evaluated. The paper is very well written.

Weaknesses: The evaluation results show that proposed method produces accurate graphs mainly for weak privacy guarantees (large \epsilon).

Comments to Authors: Without perturbation, the output of a query is the dk-2 distribution. This hides some information about the original graph, which the presented approach ignores. I wonder how much information it hides? Is it possible to quantify this? and/or take it into account in the anonymization procedure?

I understand that in the context of graph an edge corresponds to a database record. Should edges not be independent? Edges in real graphs are not created independently. The paper is not very clear about this point.

The paper misses perhaps the most related study (X. Dimitropoulos et al, "Graph Annotations in Modeling Complex Network Topologies" ACM Transactions on Modeling and Computer Simulation, vol. 19(4), Sep. 2009). This work distills an original graph to a 2K-series representation, creates empirical models of the 2K-series profile (unlike [30]), generates a new random 2K-series statistical profile from the modeled 2K-distributions (effectively adding noise), and synthesizes a graph. Combining the dk-series framework for graph anonymization makes sense. The original dk-series framework [30] produces synthetic graphs that are too similar to the original graph. This property is useful for privacy-preserving graph publishing.

I found the notation in the product of Theorem 1 somewhat cryptic, could not fully parse it.

The values of \epsilon for which the authors get good results in Section 5 are very high.

It is not clear which query model the paper assumes. Is the interactive query model supported?

Reviewer #2

Strengths: Important problem, and a good, first stab at the problem (where the problem is sharing graphs in a manner that is private and supports arbitrary queries, rather than a few select queries).

Weaknesses: 1. The paper assumes that edges in the graph is what needs to be private and the methods do not work if we wanted information about nodes to be private. This assumption is fundamental to the work and, worse, it is implicit. There is no discussion in the paper why this definition of privacy is an appropriate one.
2. The experiments are not stressing the scheme and should be done differently. Details below.

Comments to Authors: Let me expand on the two complaints above.

1. First, you are (implicitly) assuming that what needs to be hidden is whether an edge is present or absent in the graph. This is a good goal, but not the only (or the most private) one. For instance, another natural goal is to hide the presence/absence/properties of nodes in the graph. Your methods do not protect against attacks on node properties. To protect against node-level attacks, the neighboring graph would be one with an extra node added or deleted. It should be apparent that this change has way more sensitivity than adding or deleting an edge.
It would be good to make this point clear in your paper; I believe it to be an important distinction. This does not detract from your paper but makes it more precise; I think what you are doing is a great first step. At the same time, please discuss why you believe edge-level privacy to be an appropriate definition.

2. Second, I thought several aspects of your experiments should be done differently:
- The choices of \epsilon values are strange. What are they motivated by? My understanding is that \epsilon~10 is considered weak, but the lowest you go is 5. It is fine to study higher \epsilon values, but you need to study values for the more private end of the spectrum as well.
- I would like to see results in Sec. 5 for dK-PA as well. Does the poor performance that you show in earlier sections hinder it for the metrics in Sec. 5. After all, per measures in Sec. 3 and 4, even LDRC is orders of magnitude further away from the ideal.
- I disagree with how you summarize your experimental findings in Section 5.3. You basically blame dK-2 series for most of the loss in fidelity that you observe. But it is pretty clear from Figure 9, and to some extent Figure 8, that dK is not the fundamental bottleneck. dK-2 series itself is pretty close, but end-to-end measures come close only with very high \epsilon values (100). It means that it is the perturbation process that is responsible.

That said, I do like the fact that in your experiments you study applications that are not perfect matches for LDRC. You should just draw the right conclusions from them.

How do you compute the sub-series in your scheme? What if there are multiple partitions that satisfy your constraints?

Why do you see a non-monotonic behavior with respect to \epsilon in Fig. 6a? The error from the original is greater for \epsilon=10 than \epsilon=100. If this is entirely due to randomization, you should be conducting multiple experiments and plotting error bars.

Reflecting on DRC at a higher-level, I find it intriguing on two counts. First, the trick of partitioning the data and independently adding noise seems general and might find use in other places. You may consider highlighting this aspect. Second, it is also a bit counter-intuitive to me. I would have thought that considering the entire series would let one add lower noise than adding noise to individual subsets. This was essentially the observation in [39]. But you are going in the opposite direction. The difference perhaps is that you are able to find "natural" partitions of data. But it also suggests to me that better perturbation mechanisms can be designed, which treat the whole series together.

Reviewer #3

Strengths: Interesting idea. Seems fairly complete.

Weaknesses: Impact on end-to-end metrics is not clear. Unclear whether the contributions are only theoretical. Some sloppiness in the techniques. The authors show that repeat queries cannot be handled very well with diffPriv. It is unclear that distributing graphs with noise fits the diffPriv use case, which is more about answering some queries than shipping the dataset.

Comments to Authors: There is a mismatch between what differential privacy provides - the ability to respond to queries without revealing anonymity versus the canonical use case with measurement datasets which involves releasing the dataset. I did not see a way of reconciling that in this paper.

DiffPriv also has a problem dealing with succession of queries. They typically operate with a privacy budget, each answered query uses up part of it. Once budget is depleted, no more queries can be supported on the dataset without leaking information. How does the use case here work around this concern?

I see that theoretically the amount of noise injected is lower O(d_max^4/\epsilon^2) to about O(d_max^3 / \epsilon^2) but I do not see the impact on stats that matter. For example, the results in Section 5 do not compare with dk-PA, the technique that injects noise as per standard diffPriv. For Fig. 4, I cannot really tell why 'distance' is a good metric. Employ something that users may care about, and show how the accuracy of that maps onto 'distance'.

The required amount of privacy parameter, \epsilon, impacts results. \epsilon=100 is great, others not so good. In Fig. 5, the distributions are far even when the X axes is on a log scale. How does a user choose \epsilon? How exactly is an \epsilon = 5 better in terms of privacy than \epsilon = 100? The loss in accuracy is clear but the gains in terms of privacy not.

What about the AS graph makes it much harder to keep accurate? for e.g., at \epsilon=100, the accuracy is quite poor.

dK-Graph model: For network topology aspects, this is a good model. Is there nothing in the social graph that goes beyond topology of the friend graph? Metadata? Group memberships? Activity? (When is she online more often? Does he play farmville?) Correlation in activity?

Theorem 1 seems more of a sufficient than a necessary condition.

\sigma is not E[Lap(...)] which by definition is zero.

Error analysis, the lower bound seems impossible, i.e., the minimum value of 1.

It is unclear if the L2 minimization works across partitions, as it does in the vanilla case.

Sec. 5: it is not clear until the end of Sec. 5.1 that by dK you mean dK-2, which loses a lot of fidelity (Fig. 7). Use dK-3 instead?

Reviewer #4

Strengths: Anonymization is a useful tool. We lack as many datasets in this domain as we would like because commercial interests prevent them being shared.

Weaknesses: The obsession of graph theorists with node degree is really damaging to networking research. It is not clear that node degree sequences really tell us anything we need to know about data networks, or that the node-degree sequence approach really does what it claims to do.

Comments to Authors: A lot of this paper is based on statements in [30]. I never found that that paper convincingly showed that the ensemble of graphs it generated were (i) meaningfully different from the graphs that they came from (two graphs that differ by two edits can easily be homomorphic - are they then usefully different); and (ii) showed that the resulting graphs were similar in any sense except those related to node degree.

It may naively seem that both problems cannot be true, because they are almost the opposite problem. However, the problems can be different for different instantiations. For instance, for either the clique, or graph with no edges, all possible graphs with identical node degree distribution are the same. On the other hand, we know from the work of Willinger et al that graphs with identical degree distributions can be quite different in nature. Adding higher order degree sequences does not fix this. So we could have a mapping that for some graphs just creates what are effectively homomorphisms, and for others, creates a range of graphs that are in no useful sense similar to the original.

What is the worst case for this algorithm in both senses? Does it fail to anonymise for some cases, and fail to produce meaningful information in others? Limiting to dK-2 introduces similar questions. Adding noise complicates the matter further.

The graph datasets lack the types of labels that would make them interesting for many practical problems, e.g., link capacity, policies and so on. Most of the graph datasets have substantial errors. What is the effect of errors in the initial measurements?

As the framework is defined in terms of a particular set of queries, why not just distribute the results of the query on the graph? We do not know that the technique will work for other queries, so we

cannot just blindly apply them when we come to another graph. If the dK sequence is so important, why generate alternative graphs at all: just give people the dK sequence and let them work with it.

Reviewer #5

Strengths: - Addresses an important and difficult problem with some novel angle and non-trivial observation.
- combines design with concrete validation that provides more ground (even if it does not close the topic).

Weaknesses: - The reader should be ready to accept semi-proof and lack of complete rigorous formal model and proof, for the sake of exposition of interesting observations.
- Limitations are severe (only releasing k=2 reduces the k series to its single most expression: assortativity). The extension is conceptually interesting but far from being as attainable as the authors say.
- The levels of privacy proposed are not ready for prime time. Most \epsilon-differential privacy work recommends \epsilon at most 0.1 this paper considers the parameter between 5 and 100!

Comments to Authors: I think this is an interesting candidate for publication. The use of dk series seems promising to address the need to manipulate realistic graphs, this paper is making one step for it. This has potential to impact an important problem.

We are far from a contribution solidly written in the marble given the choices made by the authors in terms of high \epsilon and a rather poor exposition of the methods, which seem to follow more high-level intuition than precise first principles. Nevertheless the point that partitioning improves sensitivity and brings guarantee is a good catch, and the paper makes the case serious.

p.1 “We take a different approach to address the above question, by making the observation” the current claim of ownership of this observation is strange. This tension is the very much at the core of much research including k-anonymity, differential privacy.

The limitation of k=2 might appear a bit hard to swallow after such a grand claim. I still think it is an interesting step because that is the way to go. Who knows how sensitivity behave for k=3 and if any partitioning make sense.

“In contrast, our goal is to inject changes into all aspects of the graph topology, instead of focusing on a single graph metric.” You do not reproduce all aspects, and you will likely never. You propose to recover what the dk series can obtain, starting with k=2. It does already make your approach original (and promising) but stating it so broadly is a countersense.

“Unfortunately, the author asserts there are incorrect results in the paper 1.” This is perhaps unfortunate but it does not explain how your method is intrinsically better. It would much stronger to highlight the difference first and then mention this point.

Lemma 1 is a partial result. You only provide an upper bound, which does not prove that the real sensitivity is necessarily high.

The statement of error measure is very vague. How does random noise alter the actual structure of the graph?

Please clarify what happens between clusters? Are the data lost with some forms of random generation of links between them?

Have you ever seen a single paper advocating \epsilon between 5 and 100? You are essentially saying the users “Do not worry, your chance of being identified by joining the database are only multiplied between 148 and about 10^43”. What kind of guarantee is that?

Response from the Authors

We thank the reviewers for their insightful comments. Several comments were results of ambiguous text in the paper, which we have addressed by clarifying our claims and assumptions and providing deeper explanations of our findings. In particular, we explain that the omission of the dK-PA was simply because it generated so much noise that the dK-generator failed to generate matching graphs. Two additional key points stood out in the comments, and we address them in detail below.

First, on the issue of dK-2 as a graph statistical representation, we modified text to more clearly explain the advantages and the limitations of our choice. We explain that we require a statistical representation of a graph that can be converted to and from an unique graph. The dK-series is ideal for this. We use the dK-2 series, because it is the most detailed dK-series that has a corresponding graph generator (e.g. there is currently no known dK-3 series graph generator that works on large graphs). While the choice of dK-2 limits the accuracy of our current model, our methodology is general, and can be used with higher order dK-series when their generators are discovered (e.g. we are currently working on developing a scalable dK-3 generator). It is possible that providing privacy on higher order dK-series may require more severe noise, which could consequently destroy their higher accuracy. Therefore, our conclusion is that higher order dK-series will become a practical solution only if we are able to preserve their accuracy through the perturbation process and when a generator will be invented.

Second, we address via text edits questions on the choice of \epsilon: smaller \epsilon indicates stronger privacy. We use moderate to high values of \epsilon in our tests for two reasons. One, we wanted to find the \epsilon value that contributes to the smallest noise such that it produces a graph statistically similar to the synthetic dK-2 graph with no privacy. Thus we can indirectly quantify the level of privacy inherent in a synthetic graph without additional privacy constraints. We show that this property is achieved when \epsilon is equal to 100. In addition, the dK-2 series is a very sensitive function and naturally requires high level of noise to guarantee strong privacy. Our primary goal was to identify the feasibility of this approach, and leave further optimizations to achieve high fidelity graphs for lower \epsilon values as goals for future work.