Sharing Social Networks Using a Novel Differentially Private Graph Model

Tianchong Gao*, Feng Li*
*Indiana University-Purdue University Indianapolis, Indianapolis, IN, U.S.A.
tgao@iupui.edu, fengli@iupui.edu

***Abstract*—Online social networks (OSNs) often contain sensitive information about individuals. Therefore, anonymizing social network data before releasing it becomes an important issue. Recent research introduces several graph abstraction models to extract graph features and add sufficient noise to achieve differential privacy.**

In this paper, we design and analyze a comprehensive differentially private graph model that combines the dK-1, dK-2, and dK-3 series together. The dK-1 series stores the degree frequency, the dK-2 series adds the joint degree frequency, and the dK-3 series contains the linking information between edges. In our scheme, low dimensional data makes the regeneration process more executable and effective, while high dimensional data preserves additional utility of the graph. As the higher dimensional model is more sensitive to the noise, we carefully design the executing sequence. The final released graph increases the graph utility under differential privacy.

***Index terms*—Social network data publishing; anonymization; differential privacy; dK graph abstraction model**

I. Introduction

Studying OSNs through graph analysis could produce knowledge of human social relationships, help feed advertisements to recommendation targets, and evaluate the effectiveness of applications. Since OSN data contains personal information, any releasing procedure without sufficient anonymization work causes panic to the users of social media. Various anonymization techniques have been proposed. Differential privacy is one of the most remarkable techniques, since it could theoretically achieve a strong privacy guarantee [2].

Differential privacy requires graph abstraction models to convert the graph structure into numerical-like data. Fig. 1 gives an example of the dK model. The dK model is separated into different dimensions. The dK-N model captures the degree distribution of connected components of size N. For example, $\langle 1, 4\rangle = 2$ means that there are two node groups of degree 1 and degree 4. *Sala et al.* employed the dK-2 series as the graph abstraction model to achieve differential privacy [10].

However, deploying one abstraction model can only capture some aspects of information, while other utilities are lost in the published graph. For example, because the dK-2 graph model is the record of edges, it may not preserve information involving more than two nodes, e.g., the clustering coefficient. The limitations in the models restrict their ability to achieve structural similarity under differential privacy. Therefore, choosing the right abstraction model becomes an important issue. Mahadevan et al. proved that dK models in higher dimensions have more information than the ones in lower dimensions, e.g., the dK-3 model is more precise than the dK-2 model [5].

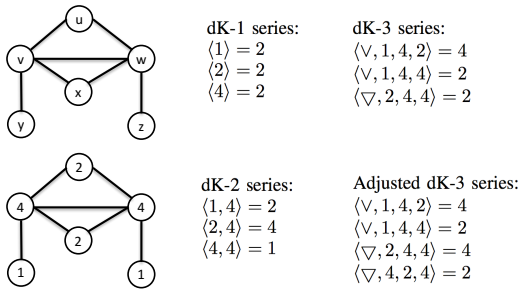

Fig. 1: An example of the dK model

Our initial idea is to preserve differential privacy with the dK-3 model. To the best of our knowledge, there is no systematic regeneration algorithm for dK-3 model because of its complexity [5, 10]. In our study, we also find that it is hard to reconstruct the graph with only the dK-3 series. However, after studying the differences between the dK-1, dK-2, and dK-3 series, we find that low dimension dK series, i.e., dK-1 and dK-2, can help the regeneration process. And we can use some rewiring algorithms to inject the dK-3 series in our published graph.

In this paper, we absorb the benefits of different models and design a new comprehensive model that combines three levels of dK graph models together. To achieve differential privacy, we introduce noise on the dK-2 level, which causes less distortion than on dK-3 level. Then we use the perturbed dK-2 series to get the corresponding dK-3 and dK-1 series. After that, we use three levels of dK series together in our scheme to construct a new graph. The noise impact is the major challenge in the graph regeneration process. Although the three models in our scheme are closely related, they may conflict with each other because of noise. Hence, we first use some dK information to regenerate an intermediate graph, then use the remaining information to rewire the edges.

The major technical contributions are the following: (1) We are the first to build the systematic regeneration algorithm for embedding dK-3 information in graph anonymization, which helps to preserve more utility than existing dK models. (2) We combine the dK-3 model with both dK-1 and dK-2 models in sampling and graph regeneration, which mitigates the high sensitivity and complexity in the dK-3 model and makes the design practical. (3) We design two different routes, CAT and LTH, to generate the graph efficiently and effectively, even under the noise impact. (4) We reveal the insights and challenges of using different levels of dK abstraction models jointly to enhance the utility under differential privacy.

This is the author's manuscript of the article published in final edited form as:

Gao, T., & Li, F. (2019). Sharing Social Networks Using a Novel Differentially Private Graph Model. 2019 16th IEEE Annual Consumer Communications Networking Conference (CCNC), 1–4. https://doi.org/10.1109/CCNC.2019.8651689

II. PRELIMINARIES

A. The dK graph model

In this paper, an OSN graph is modeled as an undirected graph $G = (V, E)$, where V is the set of vertices and E is the set of edges. $|V|$ means the cardinality of set V. d_v is the degree of the vertex v. $e_{u,v}$ means an edge between nodes u and v.

Since differential privacy is applied on the query result, the dK graph model is chosen to transform an input graph into a set of structural statistics. Although many models can give graph statistical information, the dK graph model is better than most of them because the dK series could be used to construct a new graph having structural similarities with the original graph.

The dK-N model captures the degree distribution of connected components of size N in a target graph [5]. For example, dK-1, also known as the node degree distribution, counts the number of nodes in each degree value. The dK-2 model, also called joint degree distribution, captures the number of edges in each combination of two degree values. In this paper, we define the dimension of dK information as the subgraph size (N). Hence, the dK-1 series has lower dimension than dK-2.

The dK-3 model captures the number of 3-node subgraphs with different combinations of node degrees. Specifically, there are two kinds of 3-node subgraphs: wedges and triangles.

The wedge dK-3 entry: The dK-3 entry $\langle \vee, d_u, d_v, d_w \rangle = k$ means that there are k 3-node wedges which have the node degree values equal to d_u, d_v, and d_w, and each of the two subgraphs have at least one different node. In order to prevent double counting, d_u should be less than or equal to d_w. Assume the combination of nodes u, v and w forms such a subgraph, then w should not be the neighbor of u.

The triangle dK-3 entry: The dK-3 entry $\langle \bigtriangledown, d_u, d_v, d_w \rangle = k$ means that there are k triangles with node degree d_u , d_v, and d_w. To prevent double counting, we have $d_u \leqslant d_v \leqslant d_w$. The node set of the subgraph should be $V = \{u\} \cup \{v | e_{u,v} \in E\} \cup \{w | e_{v,w} \in E \wedge e_{u,w} \in E\}$.

The error between two dK-3 series is defined as the sum of all absolute differences in each corresponding dK-3 entry.

$$err_3 = \sum_{\text{dK-3 } entry} |k_i - k_i'|. \tag{1}$$

Similarly, err_1 and err_2 measure the errors in the dK-1 and dK-2 series. Our purpose here is to reduce these errors to decrease unnecessary noise.

III. SCHEME

Given an OSN, our goal is to publish an anonymized network that preserves the structural utility as much as possible while satisfying ϵ-differential privacy. The general idea is to add sufficient noise to the dK model and reconstruct a graph G based on the perturbed dK series.

As mentioned in previous research, a model of higher dimension is more precise, but it is difficult to directly reconstruct the graph from the dK-3 series [5, 10]. It is not a good idea to start with adding noise to the dK-3 series. Another option is to add noise to the dK-1 series. However, the dK-2 and dK-3 series also need the corresponding perturbation in order to maintain consistency. Because the dK-1 series has no information of edges, it is hard to do those perturbations. In the scheme, we inject noise into the dK-2 series.

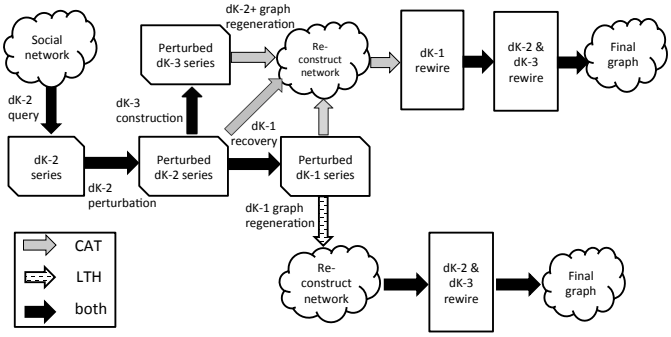

Fig. 2: Scheme overview

After injecting noise, our purpose in the graph regeneration process is to publish a graph with similar dK series as the perturbed results (in all three levels). There are two main routes in graph regeneration, starting with the dK-1 series or starting with the dK-2 series. We design two sub-schemes, shown in Fig.2, called CAT and LTH, and the mutual steps are marked in 'both'. After the regeneration part, both sub-schemes have an active rewiring procedure to mitigate their errors, e.g., the dK-2 and dK-3 series have not been used by LTH.

In the following sections, we discuss these components, which are also shown in Fig. 2:

1. Perturb the dK-2 series under differential privacy,
2. Build the dK-3 model with perturbed dK-2 series,
3. Recover the dK-1 information,
4. Reconstruct the perturbed graph with different combinations of dK series,

A. dK-2 perturbation

After counting the dK-2 entries, we apply the Laplace mechanism to achieve differential privacy [3].

Theorem 1 (LAPLACE MECHANISM)**.** *For a function $f : \mathcal{D} \to \mathbb{R}^d$, the randomized algorithm $\mathcal{A}$*

$$\mathcal{A}(G) = f(G) + Lap(\frac{\triangle f}{\epsilon}) \tag{2}$$

achieves ϵ-differential privacy [6]. The sensitivity $\triangle f$ of a function f is the maximum distance of any two neighbor databases D_1 and D_2 in ℓ_1 norm.

According to Equation 2, the noise level is determined by the sensitivity $\triangle f$ and the privacy parameter ϵ. In OSN anonymization problem, the sensitivity shows the impact of adding or deleting an edge in the model. The sensitivity of dK-2 series is shown below.

Property 1. *Given an entry $\langle d_x, d_y \rangle$ in the dK-2 model, the sensitivity $\triangle f$ is upper bounded by $2 \cdot d_x + 2 \cdot d_y + 1$.*

Proof. Let $e_{x,y}$ be a new edge added to the graph G between nodes x and y. There is one new dK-2 series $\langle d_x, d_y \rangle$ getting

incremented by 1. Also, the degrees of x and y increase from d_x and d_y to d_x+1 and d_y+1, respectively. In the original dK-2 model, there are d_x series related with the node x, which are in the form of $\langle d_u, d_x\rangle$ and $\langle d_x, d_u\rangle$. These series are deleted, while new series, $\langle d_u, d_x+1\rangle$ and $\langle d_x+1, d_u\rangle$, are added. Hence, totally $2\cdot d_x+2\cdot d_y+1$ dK-2 series are changed when the new edge is added. □

Hence, we set $\triangle f = 2\cdot d_x + 2\cdot d_y + 1$ in our scheme. The perturbed dK-2 entry is $\langle d_x, d_y\rangle = k + Lap(\frac{\triangle f}{\epsilon})$.

B. dK-3 construction

Given the dK-2 model, we construct the dK-3 model to preserve edge linking information. Particularly, if one dK-2 entry is perturbed, its corresponding dK-3 entries are also perturbed, which leaks no edge information beyond differential privacy. Hence, we examine the influence of dK-2 perturbation on the dK-3 model in the example of one edge $e_{u,v}$, then do the modification.

First, there is a simple case in which all three-node pairs in the graph are wedges. There are d_u-1 edges connected with the node u. Then the edge produces d_u-1 dK-3 entries in the form of $\langle\vee, d_x, d_u, d_v\rangle$ or $\langle\vee, d_v, d_u, d_x\rangle$. Similarly, it also produces d_v-1 dK-3 entries in the form of $\langle\vee, d_y, d_v, d_u\rangle$ or $\langle\vee, d_u, d_v, d_y\rangle$. Hence, there are totally d_u+d_v-2 dK-3 wedges entries produced by the edge $e_{u,v}$.

Second, we improve the case that the graph has some triangles. Adding an edge $e_{u,v}$ between node u and v, if they have a common neighbor x, the original entry $\langle\vee, d_u, d_x, d_v\rangle$ will be changed to $\langle\bigtriangledown, d_u, d_x, d_v\rangle$. However, if they do not have a common neighbor, there will be some new entries added, like the case before. Therefore, the total number of dK-3 entries containing the edge $e_{u,v}$ is also affected by the number of triangles.

Adjusted dK-3 model. We find that if we deploy some specific counting method for triangles, the wedges and triangles can be treated equally. Thus, the adjusted dK-3 model is proposed to simplify the calculation of the dK-3 series. The adjusted model is completely based on the basic dK-3 series. Using the adjusted model will not increase or decrease the ability of the dK-3 series to present or reconstruct the graph. The new model does not change the wedge entry $\langle\vee, d_u, d_v, d_w\rangle$. But if there is a triangle entry $\langle\bigtriangledown, d_u, d_v, d_w\rangle = k$, it will be replaced by three entries, $\langle\bigtriangledown, d_u, d_v, d_w\rangle = k$, $\langle\bigtriangledown, d_v, d_w, d_u\rangle = k$, and $\langle\bigtriangledown, d_w, d_u, d_v\rangle = k$. In the following sections, all dK-3 series are sampled in the adjusted dK-3 model. After deploying the adjusted dK-3 model, deleting or adding an edge $e_{u,v}$ always changes d_u+d_v-2 dK-3 entries. In the following sections, a wildcard character $*$ is used to match $\vee$ and $\bigtriangledown$. The dK-3 entry is like $\langle *, d_u, d_v, d_w\rangle$.

In the above section, the dK-2 series is perturbed for privacy. Each unit of increment or decrement in dK-2 entries could be viewed as one edge adding or deleting. Then we do corresponding modifications on the dK-3 series. Specifically, for increasing or decreasing, there are three possible changes in dK-3 entries

1) Replacement: If $\langle d_u, d_v\rangle$ decreased by one and $\langle d_u, d_w\rangle$ increased by one, the graph replaces the edge $e_{u,v}$ by $e_{u,w}$. So we pick $min(d_w, d_v)+d_u-2$ dK-3 entries and use the number d_w to replace d_v in the dK-3 entries.
2) Subtracting: For each unit of decrement in $\langle d_u, d_v\rangle$, the graph deletes the edge $e_{u,v}$. So we reduce the dK-3 entries containing $\langle d_u, d_v\rangle$ by the total value of d_u+d_v-2.
3) Adding: For each unit of increment in $\langle d_u, d_v\rangle$, the graph adds an edge $e_{u,v}$. The formation part is a little special because there is no original record of the neighbors of u or v. So we randomly pick a structure, wedge or triangle, and a degree number, d_x, in the range of $[1, d_{max}]$. Then we add the total value of d_u+d_v-2 to the dK-3 entries containing $\langle d_u, d_v, d_x\rangle$.

Algorithm 1 dK-1 graph regeneration (LTH)

Input: dK-1
Output: $G_1(V_1, E_1)$: the perturbed graph
1: $V_1 \leftarrow$ dK-1 ▶ add nodes with degree labels
2: **for** $i=1, i\leqslant|V|, i++$ **do**
3: pick a node u with degree d_i
4: **while** u is unsaturated **do**
5: **if** all nodes linked with u **then break** ▶ non-graphical
6: pick v with the highest degree among all unsaturated nodes unconnected with u, adds edge $e_{u,v}$
7: **end while**
8: **end for**
9: **return** G_1

C. dK-1 recovery

The dK-1 series is also important in the generation of the graph. Unlike the dK-3 series, it can be recovered directly from the dK-2 series. It is calculated by the following equation.

$$\langle d_v\rangle = \frac{\sum_{\text{dK-2 } entry}\langle d_u, d_v\rangle + \sum_{\text{dK-2 } entry}\langle d_v, d_u\rangle}{d_v}. \quad (3)$$

Equation 3 shows that the high dimensional data, e.g., dK-2, contains all the information of the low dimensional data, e.g., dK-1.

D. Graph regeneration

Given the target dK-2, dK-3, and dK-1 series, we need to regenerate the corresponding graph. Focusing on a different level of dK series, we propose two sub-schemes, namely CAT and LTH, with different regeneration algorithms.

The LTH scheme starts from the dK-1 series, the main reason is that dK-1 series is the base of the graph. If the degree of a node has an error, there will be large distortion on the corresponding dK-2 and dK-3 series. Hence, LTH just needs the dK-1 information and to generate a graph with the least err_1. By contrast, the CAT scheme considers the dK-2 and dK-3 series in regeneration because rewiring edges cannot guarantee to achieve the lowest err_2 and err_3. This scheme aims to reduce err_2 the most, while preserving some dK-3 information as well.

In both schemes, we call a node 'saturated' if it has as many neighbors as its label (dK-1 information), and call it 'unsaturated' otherwise. If the value of a dK entry in the graph reaches the target value, we call it 'full'.

LTH Algorithm 1 firstly sorts the degree sequence into a non-increasing order, which means $d_1 \geqslant d_2 \geqslant ... \geqslant d_{|V|}$.

Algorithm 2 dK-2+ graph regeneration (CAT)

Input: dK-1, dK-2, dK-3
Output: $G_1(V_1, E_1)$: the perturbed graph
$V_1 \leftarrow$ dK-1 ▶ add nodes with degree labels
while exists dK-2 entry not full **do**
——————beginning phase——————
randomly pick $\langle *, d_u, d_v, d_w \rangle$ not full in dK-3′
if $\langle d_u, d_v \rangle$ not full in dK2′ **then**
if exists u and v unconnected and unsaturated
if $* = \vee$, add edge $e_{u,v}$
if $* = \bigtriangledown$, add edge $e_{u,v}, e_{u,w}$
update dK-2 and dK-3 entries
else if exists u and v connected and unsaturated
▶ adding edge causes multi-edges
NeighborSwitch(u, v)
else mark $\langle d_u, d_v \rangle$ full, **continue**
▶ $\langle d_u, d_v \rangle$ cannot form an edge
else continue
end if
do Step 5-16, between v and w
——————continuing phase——————
pick $\langle *, d_v, d_w, d_x \rangle$, do Step 5-16 between w and x
end while
return G_1

Algorithm 3 NeighborSwitch(u,v)

find unsaturated node v' with degree d_v, $e_{u,v'} \notin E_1$
let z be a neighbor of v', $e_{z,v'} \in E_1$ and $e_{z,v} \notin E_1$
E_1 removes edge $e_{z,v'}$, adds edge $e_{z,v}$ and $e_{u,v'}$
increase $\langle d_u, d_v \rangle$ in dK-2′

Each number in the sequence also represents the target degree value of a corresponding node. Then, beginning from the first node with degree d_1, the algorithm links the node with d_1 nodes. These nodes are chosen from the set of nodes that are unconnected with the first node, which having highest degree values among the set. According to [1], a graph can be reconstructed with the exact dK-1 information if and only if every node v is connected to all d_v nodes in the leftmost part of the degree sequence (having the highest degree values).

CAT Algorithm 2 orderly picks dK-3 entries. For example, if it picks $\langle *, d_u, d_v, d_w \rangle$ in previous round, this round it picks $\langle *, d_v, d_w, d_x \rangle$. Then it tries to add one edge to the graph if these two nodes can pass the edge check. Here, the edge check means there are two unsaturated nodes with the correct degree, the two nodes are not connected, and the corresponding dK-2 entry is not full. After adding the edge, the corresponding dK-2 and dK-3 entries are updated. The regeneration process stops when there are no node pairs that can pass the edge check. Also, in the edge check process, it may happen that the only pair of unsaturated nodes are already connected. Simply connecting them together forms multi-edges in the graph, which is forbidden in OSNs. Algorithm 3 switches one neighbor from a saturated node to an unsaturated node with the same label.

The orderly picking is the continuing phase in Algorithm 2. However, if Algorithm 2 cannot find the continuing dK-3 series, e.g., $\langle *, d_v, d_w, d_x \rangle$, it uses the beginning phase to randomly choose a new dK-3 series, and adds two (wedge dK-3) or three (triangle dK-3) edges in the graph.

IV. Related Work

For sharing social network data, naive ID removal and K-anonymity are two widely used methods in social network data anonymization [7, 12, 13]. Recently, researchers proposed various kinds of structural-based de-anonymization attacks [4, 8, 9, 11]. They revealed the vulnerabilities of previous anonymization techniques in different angles. Differential privacy, which could achieve a strong privacy guarantee, was initially proposed to release certain data mining results, like the degree distribution and other graph patterns [2].

V. Conclusion

In this paper, we propose a uniform scheme that combines three levels of dK graph models to publish a perturbed social network. We design two different sub-schemes, CAT and LTH, to regenerate the graph and reduce the error under the differential privacy noise impact. Our two schemes have different merits in preserving graph utility. The design, analysis, and comparison also reveal more insights and challenges in using multiple levels of graph abstraction models together in differential private graph releasing for OSNs.

References

[1] Charo I Del Genio, Hyunju Kim, Zoltán Toroczkai, and Kevin E Bassler. Efficient and exact sampling of simple graphs with given arbitrary degree sequence. *PloS one*, 5(4):e10012, 2010.

[2] Cynthia Dwork. Differential privacy. In *Encyclopedia of Cryptography and Security*, pages 338–340. Springer, 2011.

[3] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[4] Shouling Ji, Weiqing Li, Shukun Yang, Prateek Mittal, and Raheem Beyah. On the relative de-anonymizability of graph data: Quantification and evaluation. 2016.

[5] Priya Mahadevan, Dmitri Krioukov, Kevin Fall, and Amin Vahdat. Systematic topology analysis and generation using degree correlations. In *ACM SIGCOMM Computer Communication Review*, volume 36, pages 135–146. ACM, 2006.

[6] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*, pages 94–103. IEEE, 2007.

[7] Arvind Narayanan and Vitaly Shmatikov. De-anonymizing social networks. In *Security and Privacy, 2009 30th IEEE Symposium on*, pages 173–187. IEEE, 2009.

[8] Wei Peng, Feng Li, Xukai Zou, and Jie Wu. Seed and grow: An attack against anonymized social networks. In *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2012 9th Annual IEEE Communications Society Conference on*, pages 587–595. IEEE, 2012.

[9] Jianwei Qian, Xiang-Yang Li, Chunhong Zhang, and Linlin Chen. De-anonymizing social networks and inferring private attributes using knowledge graphs. 2016.

[10] Alessandra Sala, Xiaohan Zhao, Christo Wilson, Haitao Zheng, and Ben Y Zhao. Sharing graphs using differentially private graph models. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 81–98. ACM, 2011.

[11] Guojun Wang, Qin Liu, Feng Li, Shuhui Yang, and Jie Wu. Outsourcing privacy-preserving social networks to a cloud. In *INFOCOM, 2013 Proceedings IEEE*, pages 2886–2894. IEEE, 2013.

[12] Bin Zhou and Jian Pei. Preserving privacy in social networks against neighborhood attacks. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 506–515. IEEE, 2008.

[13] Lei Zou, Lei Chen, and M Tamer Özsu. K-automorphism: A general framework for privacy preserving network publication. *Proceedings of the VLDB Endowment*, 2(1):946–957, 2009.