This is the Pre-Published Version.
The following publication Q. Ye, H. Hu, M. H. Au, X. Meng and X. Xiao, "Towards Locally Differentially Private Generic Graph Metric Estimation," 2020 IEEE 36th International Conference on Data Engineering (ICDE), Dallas, TX, USA, 2020, pp. 1922-1925 is available at https://dx.doi.org/10.1109/ICDE48307.2020.00204.

# Towards Locally Differentially Private Generic Graph Metric Estimation

Qingqing Ye*, Haibo Hu†, Man Ho Au†, Xiaofeng Meng*, Xiaokui Xiao‡
*Renmin University of China; †Hong Kong Polytechnic University; ‡National University of Singapore
*yeqq@ruc.edu.cn; haibo.hu@polyu.edu.hk; csallen@comp.polyu.edu.hk; xfmeng@ruc.edu.cn; xkxiao@nus.edu.sg*

***Abstract*—Local differential privacy (LDP) is an emerging technique for privacy-preserving data collection without a trusted collector. Despite its strong privacy guarantee, LDP cannot be easily applied to real-world graph analysis tasks such as community detection and centrality analysis due to its high implementation complexity and low data utility. In this paper, we address these two issues by presenting LF-GDPR, the first LDP-enabled graph metric estimation framework for graph analysis. It collects two atomic graph metrics — the adjacency bit vector and node degree — from each node locally. LF-GDPR simplifies the job of implementing LDP-related steps (e.g., local perturbation, aggregation and calibration) for a graph metric estimation task by providing either a complete or a parameterized algorithm for each step.**

***Index Terms*—Local differential privacy; Graph metric; Privacy-preserving graph analysis**

## I. Introduction

With the prevalence of big data and machine learning, graph analytics has received great attention and nurtured numerous applications in web, social network, transportation, and knowledge base. However, recent privacy incidents, particularly the Facebook privacy scandal, pose real-life threats to any **centralized** party who needs to safeguard graph data of individuals while providing graph analysis service to third parties. In that scandal, Facebook exposed the personal profiles of 87 million users to Cambridge Analytica through Facebook API for third-party apps [11]. The main cause is that Facebook allows these apps to access the **friends list** of a user, which helps to propagate these apps easily through friends. Unfortunately, most existing privacy models assume that the trusted party cannot be compromised, which is seldom true in practice as echoed by this scandal. With General Data Protection Regulation (GDPR) enforced in EU since May 2018, there is a compelling need to find alternative privacy models without such a trusted party.

A promising model is local differential privacy (LDP) [1], [15], where each individual user **locally perturbs her share of graph metrics** (e.g., node degree and adjacency list, depending on the graph analysis task) before sending them to the data collector for analysis. As such, the data collector does not need to be trusted. A recent work *LDPGen* [10] has also shown the potential of LDP for graph analytics. In that work, LDP is used to collect node degree for synthetic graph generation. However, such solution is usually task specific — for different tasks, such as centrality analysis and community detection, dedicated LDP solutions must be designed from scratch. To show how complicated it is, an LDP solution usually takes four steps: (1) selecting graph metrics to collect from users for the target metric (e.g., clustering coefficient, modularity, or centrality) of this task, (2) designing a local perturbation algorithm for users to report these metrics under LDP, (3) designing a collector-side aggregation algorithm to estimate the target metric based on the perturbed data, (4) designing an optional calibration algorithm for the target metric if the estimation is biased. Obviously, working out such a solution **requires in-depth knowledge of LDP**, which hinders the embrace of LDP by more graph applications.

In this paper, we address this challenge by presenting LF-GDPR (Local Framework for Graph with Differentially Private Release), the first LDP-enabled graph metric estimation framework for general graph analysis. It simplifies the job of a graph application to design an LDP solution for a graph metric estimation task by providing complete or parameterized algorithms for steps (2)-(4) as above. As long as the target graph metric can be derived from the two atomic metrics, namely, the adjacency bit vector and node degree, the parameterized algorithms in steps (2)-(4) can be completed with ease. To summarize, our main contributions of this paper are as follows.

- This is the first LDP-enabled graph metric estimation framework for a variety of graph analysis tasks.
- We present efficient perturbation algorithms on adjacency bit vector and node degree, respectively, to address data correlation among nodes.
- We provide a complete solution for local perturbation, collector-side aggregation, and calibration.

The rest of the paper is organized as follows. Section II introduces preliminaries on local differential privacy and graph analytics. Section III presents an overview of LF-GDPR. Section IV describes the implementation details of this framework. Section V draws a conclusion with future work.

## II. Local Differential Privacy on Graphs

In this paper, a graph $G$ is defined as $G = (V, E)$, where $V = \{1, 2, ..., n\}$ is the set of nodes, and $E \subseteq V \times V$ is the set of edges. For the node $i$, $d_i$ denotes its degree and $\boldsymbol{B}_i = \{b_1, b_2, ..., b_n\}$ denotes its *adjacency bit vector*, where $b_j = 1$ if and only if edge $(i, j) \in E$, and otherwise $b_j = 0$. The adjacency bit vectors of all nodes constitute the *adjacency matrix* of graph $G$, or formally, $\boldsymbol{M}_{n\times n} = \{\boldsymbol{B}_1, \boldsymbol{B}_2, ..., \boldsymbol{B}_n\}$.

Local differential privacy (LDP) [1] is proposed to assume each individual is responsible for her own tuple in the database. In LDP, each user locally perturbs her tuple using a randomized algorithm before sending it to the untrusted data collector. Formally, a randomized algorithm $\mathcal{A}$ satisfies $\epsilon$-local differential privacy, if for any two input tuples $t$ and $t'$ and for any output $t^*$, $\frac{\Pr[\mathcal{A}(\mathrm{t})=\mathrm{t}^*]}{\Pr[\mathcal{A}(\mathrm{t}')=\mathrm{t}^*]} \leq e^{\epsilon}$ holds. As with existing LDP works, we concern attacks where an adversary can infer with high confidence whether an edge exists or not, which compromises a user's relation anonymity in a social network. This directly leads to Definition 2.1.

*Definition 2.1: (Edge local differential privacy).* A randomized algorithm $\mathcal{A}$ satisfies $\epsilon$-edge local differential privacy (a.k.a., $\epsilon$-edge LDP), if and only if for any two adjacency bit vectors $\boldsymbol{B}$ and $\boldsymbol{B}'$ that differ only in one bit, and any output $s \in range(\mathcal{A})$, $\frac{\Pr[\mathcal{A}(\boldsymbol{B})=s]}{\Pr[\mathcal{A}(\boldsymbol{B}')=s]} \leq e^{\epsilon}$ holds.

## III. LF-GDPR: Framework Overview

### A. Design Principle

The core of privacy-preserving graph analytics often involves **estimating some target graph metric** without accessing the original graph. Under the DP/LDP privacy model, there are two solution paradigms, namely, generating a synthetic graph to calculate this metric [5], [7], [10] and designing a dedicated DP/LDP solution for such metric [4], [6], [9]. The former provides a general solution but suffers from low estimation accuracy as **the neighborhood information in the original graph is missing** from the synthetic graph. The latter can achieve higher estimation accuracy but cannot generalize such a dedicated solution to other problems — it works poorly or even no longer works if the target graph metric or graph type (e.g., undirected graph, attributed graph, and DAG) is changed [5].

LF-GDPR is our answer to both solution generality and estimation accuracy under the LDP model. It collects from each node $i$ two atomic graph metrics that can derive a wide range of common metrics. The first is the **adjacency bit vector** $\boldsymbol{B}$, where each element $j$ is 1 only if $j$ is a neighbor of $i$. $\boldsymbol{B}$ of all nodes collectively constitutes the adjacency matrix $\boldsymbol{M}$ of the graph. The second metric is **node degree** $d$, which is frequently used in graph analytics to measure the density of connectivity [4]. Table I lists some of the most popular graph analysis tasks in the literature [3], [8], [13] and their graph metrics, all of which can be derived from $\boldsymbol{B}$, $\boldsymbol{M}$ and $d$.

Intuitively, $d$ can be estimated from $\boldsymbol{B}$. However, given a large graph and limited privacy budget, the estimation accuracy could be too noisy to be meaningful. To illustrate this, let us assume each bit of the adjacency bit vector $\boldsymbol{B}$ is perturbed independently by the classic Randomized Response (RR) [12] algorithm with privacy budget $\epsilon$. As stated in [12], the variance of the estimated node degree $\widetilde{d}$ is

$$Var[\widetilde{d}] = n \cdot \Big[\frac{1}{16(\frac{e^{\epsilon}}{e^{\epsilon}+1} - \frac{1}{2})^2} - (\frac{d}{n} - \frac{1}{2})^2\Big] \qquad (1)$$

Even for a moderate social graph with extremely large privacy budget, for example, $d = 100$, $n = 1M$, and $\epsilon = 8$ (the largest $\epsilon$ used in [10] is 7), $Var[\widetilde{d}] \approx 435 > 4d$, which means the variance of the estimated degree is over 4 times that of the degree itself. As such, we choose to spend some privacy budget on an independently perturbed degree. This further motivates us to design an optimal privacy budget allocation between adjacency bit vector $\boldsymbol{B}$ and node degree $d$, to minimize the distance between the target graph metric and the estimated one.

TABLE I
Popular graph analysis tasks and metrics

| Graph Analysis Task | Graph Metric Concerned | Derivation from $\boldsymbol{B}$, $\boldsymbol{M}$, and $d$ |
|---|---|---|
| synthetic graph generation | clustering coefficient | $cc_i = \frac{\boldsymbol{M}^3_{ii}}{d_i(d_i-1)}$ |
| community detection, graph clustering | modularity | $Q_c = \frac{\|\|\boldsymbol{M}_c\|\|}{\sum d} - \frac{\|\|\boldsymbol{d}_c\|\|^2}{(\sum d)^2}$ |
| node role, page rank | degree centrality | $c_i = d_i$ |
| | eigenvector centrality | $c_i = \boldsymbol{B}_i\boldsymbol{M}^k$ |
| connectivity analysis (clique / hub) | structural similarity | $\tau(i,j) = \frac{\|\|\boldsymbol{B}_i \bigcap \boldsymbol{B}_j\|\|}{\sqrt{d_i d_j}}$ |
| node similarity search | cosine similarity | $\tau(i,j) = \frac{\boldsymbol{B}_i\boldsymbol{B}'_j}{\sqrt{d_i d_j}}$ |

To summarize, in LF-GDPR each node sends two perturbed atomic metrics, namely, the adjacency bit vector $\widetilde{\boldsymbol{B}}$ (perturbed from $\boldsymbol{B}$) and node degree $\widetilde{d}$ (perturbed from $d$), to the data collector, who then aggregates them to estimate the target graph metric.

### B. LF-GDPR Overview

LF-GDPR works as shown in Fig. 1. A data collector who wishes to estimate a target graph metric $F$ first reduces it from the adjacency matrix $\boldsymbol{M}$ and degree vector $\boldsymbol{d}$ of all nodes by deriving a mapping function $F = Map(\boldsymbol{M}, \boldsymbol{d})$ (step ①). Based on this reduction, LF-GDPR allocates the total privacy budget $\epsilon$ between $\boldsymbol{M}$ and $\boldsymbol{d}$, denoted by $\epsilon_1$ and $\epsilon_2$, respectively (step ②). Then each node locally perturbs its adjacency bit vector $\boldsymbol{B}$ into $\widetilde{\boldsymbol{B}}$ to satisfy $\epsilon_1$-edge LDP, and perturbs its node degree $d$ into $\widetilde{d}$ to satisfy $\epsilon_2$-edge LDP (step ③). According to the composability of LDP, each node then satisfies $\epsilon$-edge LDP. Note that this step is challenging as both $\boldsymbol{B}$ and $d$ are correlated among nodes. For $\boldsymbol{B}$, the $j$-th bit of node $i$'s adjacency bit vector is the same as the $i$-th bit of node $j$'s adjacency bit vector. For $d$, whether $i$ and $j$ has an edge affects both degrees of $i$ and $j$. Sections IV-B and IV-C solve this issue and send out the perturbed $\boldsymbol{B}$ and $d$, i.e., $\widetilde{\boldsymbol{B}}$ and $\widetilde{d}$. The data collector receives them from all nodes, aggregates them according to the mapping function $Map(\cdot)$ to obtain the estimated target metric $\widetilde{F}$, and further calibrates it to suppress estimation bias and improve accuracy (step ④). The resulted $\widetilde{F}$ is then used for graph analysis. The detailed implementation of LF-GDPR for steps ①③④ will be presented in Section IV. Note that the algorithms in steps ①②④ are parameterized, which can only be determined when the target graph metric $F$ is specified.

*Example III-B.* **LF-GDPR against Facebook Privacy Scandal.** Facebook API essentially controls how a third-party app accesses the data of each individual user. To limit the

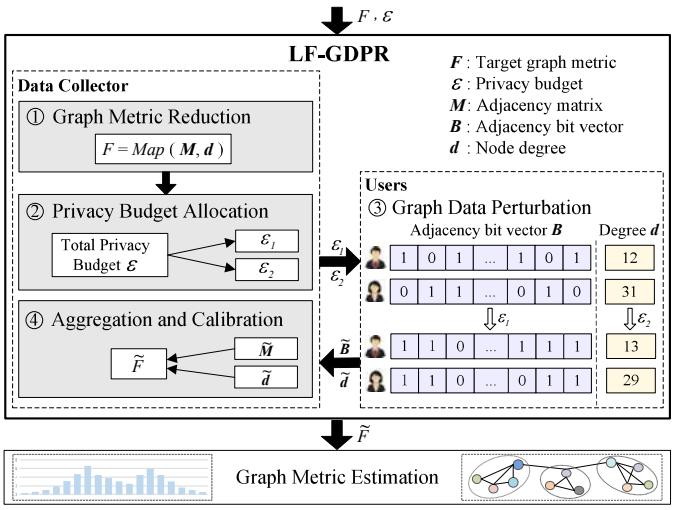



Fig. 1. An overview of LF-GDPR

access right of an average app (e.g., the one developed by Cambridge Analytica) while still supporting graph analytics, Facebook API should have a new permission rule that only allows such app to access the perturbed adjacency bit vector and degree of a user's friends list under $\epsilon_1$ and $\epsilon_2$-edge LDP, respectively. In the Cambridge Analytica case, the app is a personality test, so the app developer may choose structural similarity as the target graph metric and use the estimated value for the personality test. To estimate structural similarity, the app then implements steps ①②④ of LF-GDPR. On the user side, each user $u$ has a privacy budget $\epsilon_u$ for her friends list. If $\epsilon_u \geq \epsilon_1 + \epsilon_2$, the user can grant access to this app for perturbed adjacency bit vector and degree; otherwise, the user simply ignores this access request.

## IV. LF-GDPR: Implementation

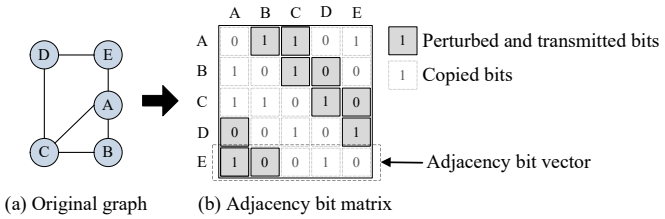
In this section, we present the implementation details of LF-GDPR. We first discuss graph metric reduction (step ①), followed by the perturbation protocols for adjacency bit vector and node degree, respectively (step ③). Then we elaborate on the aggregation and calibration algorithm (step ④).

### A. Graph Metric Reduction

The reduction outputs a polynomial mapping function $Map(\cdot)$ from the target graph metric $F$ to the adjacency matrix $\boldsymbol{M} = \{\boldsymbol{B}_1, \boldsymbol{B}_2, ..., \boldsymbol{B}_n\}$ and degree vector $\boldsymbol{d} = \{d_1, d_2, ..., d_n\}$, i.e., $F = Map(\boldsymbol{M}, \boldsymbol{d})$. Without loss of generality, we assume $F$ is a polynomial of $\boldsymbol{M}$ and $\boldsymbol{d}$. That is, $F$ is a sum of terms $F_l$, each of which is a multiple of $\boldsymbol{M}$ and $\boldsymbol{d}$ of some exponents. Since $F$ and $F_l$ are scalars, in each term $F_l$, we need functions $f$ and $g$ to transform $\boldsymbol{M}$ and $\boldsymbol{d}$ with exponents to scalars, respectively. Formally,

$$F = \sum\nolimits_l F_l = \sum\nolimits_l f_{\phi_l}(\boldsymbol{M}^{k_l}) \cdot g_{\psi_l}(\boldsymbol{d}), \tag{2}$$

where $\boldsymbol{M}^{k_l}$ is the $k_l$-th power of adjacency matrix $\boldsymbol{M}$ whose cell $(i, j)$ denotes the number of paths between node $i$ and $j$ of length $k_l$, $\phi_l$ projects a matrix to a cell, a row, a column




(a) Original graph (b) Adjacency bit matrix

Fig. 2. Illustration of $RABV$ protocol

or a sub-matrix, and $f_{\phi_l}(\cdot)$ denotes an aggregation function $f$ (e.g., sum) after projection $\phi_l$. Likewise, $\psi_l$ projects a vector to a scalar or a sub-vector, and $g_{\psi_l}(\cdot)$ denotes an aggregation function $g$ after $\psi_l$.

As such, the metric reduction step is to determine $k_l$, $f_{\phi_l}(\cdot)$, and $g_{\psi_l}(\cdot)$ for each term $F_l$ in Eq. 2.

### B. Adjacency Bit Vector Perturbation

An intuitive approach, known as *Randomized Neighbor List* (*RNL*) [10], perturbs each bit of the vector independently by the classic Randomized Response (RR) [12]. Formally, given an adjacency bit vector $\boldsymbol{B} = \{b_1, b_2, ..., b_n\}$, and privacy budget $\epsilon_1$, the perturbed vector $\widetilde{\boldsymbol{B}} = \{\widetilde{b}_1, \widetilde{b}_2, ..., \widetilde{b}_n\}$ is obtained as follows:

$$\widetilde{b}_i = \begin{cases} b_i & \text{w.p. } \frac{e^{\epsilon_1}}{1+e^{\epsilon_1}} \\ 1 - b_i & \text{w.p. } \frac{1}{1+e^{\epsilon_1}} \end{cases} \tag{3}$$

*RNL* is proved to satisfy $\epsilon_1$-edge LDP for each user. However, **for undirected graphs, *RNL* can only achieve $2\epsilon_1$-edge LDP for the collector**, because the data collector witnesses the same edge perturbed twice and independently. Let $\widetilde{\boldsymbol{M}} = \{\widetilde{\boldsymbol{B}}_1, \widetilde{\boldsymbol{B}}_2, ..., \widetilde{\boldsymbol{B}}_n\}$ denote the perturbed adjacency matrix. The edge between node $i$ and $j$ appears in both $\widetilde{M}_{ij}$ and $\widetilde{M}_{ji}$, each perturbed with privacy budget $\epsilon_1$. Then according to the theorem of composability, *RNL* becomes a $2\epsilon_1$-edge LDP algorithm for an undirected graph, which is less private. Furthermore, *RNL* requires each user to perturb and send all $n$ bits in the adjacency bit vector to data collector, which incurs a high computation and communication cost.

To address the problems of *RNL*, we propose a more private and efficient protocol *Randomized Adjacency Bit Vector* ($RABV$) to perturb edges in undirected graphs. As shown in Fig. 2(b), the adjacency matrix is composed of $n$ rows, each corresponding to the adjacency bit vector of a node. For the first $1 \leq i \leq \lfloor \frac{n}{2} \rfloor$ nodes, $RABV$ uses RR as in Eq.3 to perturb and transmit $t = \lfloor \frac{n}{2} \rfloor$ bits (i.e., bits in grey) — from the $(i+1)$-th bit to the $(i+1+t \mod n)$-th bit; for the rest nodes, $RABV$ uses RR to perturb and transmit $t = \lfloor \frac{n-1}{2} \rfloor$ bits in the same way. In essence, $RABV$ **perturbs one and only one bit** for each pair of symmetric bits in the adjacency matrix. The data collector can then obtain the whole matrix by copying bits in grey to their symmetric positions.

Following the same proof of $RNL$, $RABV$ is guaranteed to satisfy $\epsilon_1$-edge LDP for the collector. Further, since each node only perturbs and transmits about half of the bits in an adjacency bit vector, $RABV$ significantly reduces computation and communication cost of *RNL*.

### C. Node Degree Perturbation

Releasing the degree of a node while satisfying edge $\epsilon$-LDP is essentially a centralized DP problem because all edges incident to this node, or equivalently, all bits in its adjacency bit vector, form a database and the degree is a count function. In the literature, *Laplace Mechanism* [2] is the predominant technique to perturb numerical function values such as counts. As such, LF-GDPR adopts it to perturb the degree $d_i$ of each node $i$. According to the definition of edge LDP, two adjacency bit vectors $\boldsymbol{B}$ and $\boldsymbol{B}'$ are two neighboring databases if they differ in only one bit. As such, the sensitivity of degree (i.e., count function) is 1, and therefore adding Laplace noise $Lap(\frac{1}{\epsilon_2})$ to the node degree can satisfy $\epsilon_2$-LDP. That is, $\widetilde{d}_i = d_i + Lap(\frac{1}{\epsilon_2})$.

Similar to perturbing adjacency bit vector, however, in the above naive approach the data collector witnesses two node degrees $d_i$ and $d_j$ perturbed independently, but they share the same edge between $i$ and $j$. As DP or LDP does not refrain an adversary from possessing any background knowledge, in the worst case the collector already knows all edges except for this one. As such, witnessing the two node degrees $d_i$ and $d_j$ is degenerated to witnessing the edge between $i$ and $j$ twice and independently.

Unfortunately, the remedy that works for perturbing adjacency bit vector cannot be adopted here, as direct bit copy is not feasible for degree. As such, we take an alternative approach to increase the Laplace noise. The following theorem proves that if we add Laplace noise $Lap(\frac{2}{\epsilon_2})$ to every node degree, $\epsilon_2$-LDP can be satisfied for the collector.

*Theorem 4.1:* A perturbation algorithm $\mathcal{A}$ satisfies $\epsilon_2$-LDP for the collector if it adds Laplace noise $Lap(\frac{2}{\epsilon_2})$ to every node degree $d_i$, i.e., $\widetilde{d}_i = \mathcal{A}(d_i) = d_i + Lap(\frac{2}{\epsilon_2})$.

PROOF. Please refer to our technical report [14]. □

### D. Aggregation and Calibration

Upon receiving the perturbed adjacency matrix $\widetilde{\boldsymbol{M}}$ and degree vector $\widetilde{\boldsymbol{d}}$,[1] the data collector can estimate the target graph metric $\widetilde{F}$ by aggregation according to Eq. 2 with a calibration function $\mathcal{R}(\cdot)$:

$$\widetilde{F} = \sum\nolimits_l \mathcal{R}\left(f_{\phi_l}(\widetilde{\boldsymbol{M}}^{k_l})\right) \cdot g_{\psi_l}(\widetilde{\boldsymbol{d}}) \tag{4}$$

The calibration function aims to suppress the aggregation bias of $\widetilde{\boldsymbol{M}}$ propagated by $f_{\phi_l}$. On the other hand, no calibration is needed for $g_{\psi_l}(\widetilde{\boldsymbol{d}})$ as $\widetilde{\boldsymbol{d}}$ is already an unbiased estimation of $\boldsymbol{d}$, thanks to the Laplace Mechanism.

To derive $\mathcal{R}(\cdot)$, we regard $\mathcal{R}$ as the mapping between $f_{\phi_l}(\boldsymbol{M}^{k_l})$ and $f_{\phi_l}(\widetilde{\boldsymbol{M}}^{k_l})$. In other words, $\mathcal{R}$ estimates $f_{\phi_l}(\boldsymbol{M}^{k_l})$ after observing $f_{\phi_l}(\widetilde{\boldsymbol{M}}^{k_l})$. Formally,

$$\mathcal{R} : f_{\phi_l}(\widetilde{\boldsymbol{M}}^{k_l}) \rightarrow f_{\phi_l}(\boldsymbol{M}^{k_l})$$

Further, the following theorem shows the accuracy guarantee of LF-GDPR.

[1]In the sequel, $\widetilde{\boldsymbol{d}}$ denotes the refined degree $\widetilde{\boldsymbol{d}}^*$ to simplify the notation.

*Theorem 4.2:* For a graph metric $F$ and our estimation $\widetilde{F}$, with at least $1-\beta$ probability, we have

$$|F - \widetilde{F}| = O(\sqrt{\mathbb{E}[\widetilde{F}^2] \cdot log(1/\beta)}\,)$$

PROOF. Please refer to our technical report [14]. □

## V. CONCLUSION

This paper presents a parameterized framework LF-GDPR for privacy-preserving graph metric estimation and analytics with local differential privacy. The building block is a user-side perturbation algorithm, and a collector-side aggregation and calibration algorithm. LF-GDPR simplifies the job of developing a practical LDP solution for a graph analysis task by providing a complete solution for all LDP steps. As for future work, we plan to extend LF-GDPR to more specific graph types, such as attributed graph and DAG. We will also evaluate the performance of LF-GDPR on other graph analysis tasks such as influential node analysis to demonstrate its wide applicability.

## ACKNOWLEDGMENT

This work was supported by National Natural Science Foundation of China (Grant No: 91646203, 61941121, 61572413, U1636205, 61532010, 91846204 and 61532016), the Research Grants Council, Hong Kong SAR, China (Grant No: 15238116, 15222118 and C1008-16G) (corresponding author: Xiaofeng Meng).

## REFERENCES

[1] J. C. Duchi, M. I. Jordan, and M. J. Wainwright. Local privacy and statistical minimax rates. In *FOCS*, pages 429–438. IEEE, 2013.
[2] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284. Springer, 2006.
[3] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 3 edition, 2011.
[4] M. Hay, C. Li, G. Miklau, and D. Jensen. Accurate estimation of the degree distribution of private networks. In *ICDM*, pages 169–178, 2009.
[5] Z. Jorgensen, T. Yu, and G. Cormode. Publishing attributed social graphs with formal privacy guarantees. In *SIGMOD*, pages 107–122, 2016.
[6] S. P. Kasiviswanathan, K. Nissim, S. Raskhodnikova, and A. Smith. Analyzing graphs with node differential privacy. In *TCC*, pages 457–476. Springer, 2013.
[7] W. Lu and G. Miklau. Exponential random graph estimation under differential privacy. In *KDD*, pages 921–930. ACM, 2014.
[8] T. Martin, X. Zhang, and M. Newman. Localization and centrality in networks. *Physical review E*, 90(5):052808, 2014.
[9] K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. In *STOC*, pages 75–84. ACM, 2007.
[10] Z. Qin, T. Yu, Y. Yang, I. Khalil, X. Xiao, and K. Ren. Generating synthetic decentralized social graphs with local differential privacy. In *CCS*, pages 425–438. ACM, 2017.
[11] B. Stephanie, Facebook Scandal a 'Game Changer' in Data Privacy Regulation. *Bloomberg*, Apr 8, 2018.
[12] S. L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.
[13] X. Xu, N. Yuruk, Z. Feng, and T. A. J. Schweiger. SCAN: a structural clustering algorithm for networks. In *KDD*, pages 824–833. ACM, 2007.
[14] Q. Ye, H. Hu, M. H. Au, X. Meng, and X. Xiao. LF-GDPR: A framework for estimating graph metrics with local differential privacy. Technical report. http://www.eie.polyu.edu.hk/%7ehaibohu/papers/lfgdpr.pdf.
[15] Q. Ye, H. Hu, X. Meng, and H. Zheng. PrivKV: Key-value data collection with local differential privacy. In *S&P*, pages 317–331. IEEE, 2019.