Hindawi
Security and Communication Networks
Volume 2017, Article ID 4267921, 10 pages
https://doi.org/10.1155/2017/4267921

Research Article

Differential Privacy for Edge Weights in Social Networks

Xiaoye Li,[1,2] Jing Yang,[1] Zhenlong Sun,[1,2] and Jianpei Zhang[1]

[1]*College of Computer Science and Technology, Harbin Engineering University, Harbin, Heilongjiang, China*
[2]*Modern Education Technology Center, Qiqihar University, Qiqihar, Heilongjiang, China*

Correspondence should be addressed to Jing Yang; yangjing@hrbeu.edu.cn

Received 3 January 2017; Revised 6 February 2017; Accepted 23 February 2017; Published 9 March 2017

Academic Editor: Alexandre Viejo

Social networks can be analyzed to discover important social issues; however, it will cause privacy disclosure in the process. The edge weights play an important role in social graphs, which are associated with sensitive information (e.g., the price of commercial trade). In the paper, we propose the MB-CI (Merging Barrels and Consistency Inference) strategy to protect weighted social graphs. By viewing the edge-weight sequence as an unattributed histogram, differential privacy for edge weights can be implemented based on the histogram. Considering that some edges have the same weight in a social network, we merge the barrels with the same count into one group to reduce the noise required. Moreover, k-indistinguishability between groups is proposed to fulfill differential privacy not to be violated, because simple merging operation may disclose some information by the magnitude of noise itself. For keeping most of the shortest paths unchanged, we do consistency inference according to original order of the sequence as an important postprocessing step. Experimental results show that the proposed approach effectively improved the accuracy and utility of the released data.

1. Introduction

Social networks, such as Facebook and Twitter, have played an important role in people's daily social interaction. Social network analysis attempts to discover important social issues, including disease transmission, emotional contagion, and occupational mobility. Due to the need of scientific research and data sharing, social networks are supposed to release data without leaking privacy information. Privacy can be guaranteed by disturbing or encrypting the original data, or doing anonymous processing before releasing the data [1–3].

Privacy is a charged term, meaning different things to different people. In social networks, the edge weights may reflect the frequency of communication, the price of commercial trade, the intimacy of relationship, and so forth, which are associated with sensitive information. A typical example is an intelligence network, in which edge weights denote the contact frequencies of two institutions. Too-frequent communications may imply potential problems. Another example is a commercial trade network, in which edge weights indicate the transaction price between two companies. Most managers would be reluctant to reveal a commercial secret to their adversaries, due to the fierce competition. Our goal is to protect the edge weights in social networks without leakage while preserving as much utility as possible.

Das et al. [4] considered edge-weight anonymization in social graphs. They built a linear programming (LP) model to preserve the properties of the graph, for example, the shortest paths, k-nearest neighbors, and minimum spanning tree, which are expressible as linear functions of the edge weights. Liu et al. [5] considered preserving the weights of some edges, while trying to preserve the shortest-path lengths and exactly the same shortest paths of some pairs of nodes. They developed two privacy-preserving strategies: Gaussian randomization multiplication and a greedy perturbation algorithm based on graph theory. Costea et al. [6] analyzed how differential privacy can be used to protect the edge weights in graph structures. Nonetheless, simply adding Laplacian noise to the edge weights would distort the accuracy very significantly. Our approach is to disturb the edge weights via differential privacy for protection, which effectively improves the accuracy and utility of the released data.

Hayy et al. [7] showed that it is possible to significantly improve the accuracy of a general class of histogram query while satisfying differential privacy. The approach carefully chooses the queries to evaluate, and then exploits the consistency constraints that should hold over the noisy output. After a postprocessing phase, the final output is differentially private and consistent, but, in addition, it is often much more accurate. The technique was used to very precisely estimate the degree sequence of a graph, which is an important instance of an unattributed histogram. Inspired by the above, we treat the edge-weight sequence as an unattributed histogram in the proposed approach, which is a key step in this paper. To better keep the shortest paths unchanged, we do consistency inference according to original order of the sequence.

Xu et al. [8, 9] proposed two algorithms, namely, NoiseFirst and StructureFirst, for computing differentially private histograms. The main difference lies in the relative order of the noise injection and the histogram construction. Going one step further, they extended both solutions to answer arbitrary range queries. StructureFirst constructs an optimal histogram based on the original data. Then, the algorithm randomly moves the boundaries between the barrels, which adds noise to the structure of the histogram. After setting down all the boundaries, Laplace noise is added to the average counts. Thus, this method introduces two kinds of errors: construction error and noise error. For the specific application, our strategy is to merge all barrels with the same count into one group and then add Laplace noise to each count, so the proposed approach only has noise error. In the step of merging barrels, to prevent leaking some information by the magnitude of the noise itself, inspired by literature [10, 11], we propose the definition of k-indistinguishability between groups to guarantee differential privacy.

2. Background

In this section, we review the definition of differential privacy and its implementation mechanism. Then, we clarify the concepts of an unattributed histogram versus a conventional histogram.

2.1. Differential Privacy. Dalenius [12] proposed an issue for statistical databases: no one should learn anything about an individual while accessing the database. Nevertheless, the type of privacy that is an absolute guarantee about disclosures cannot be achieved because of auxiliary information. Differential privacy [13] sidesteps this problem to the related ones; any given disclosure will be within a small multiplicative factor. Note that a bad disclosure may still occur, but it will not be caused by the presence of an individual's data in the database. Differential privacy can hide the influence of a single record, that is, the output probability of the same results will not change significantly, whether a record is in the data set or not. Hence, differential privacy makes no assumptions about the background knowledge of any potential adversary. However, we still face the challenge of making the tradeoff between protecting privacy information and maintaining the data utility.

Differential privacy was presented in a series of Dwork's papers [14–18] and its implementation mechanism was presented in the literature [19, 20]. McSherry [21] pointed out that a differentially private algorithm for some complex privacy problem satisfies two combination properties. Recently, differential privacy has mainly been used in data publishing, including releasing histograms [7–9, 22–24] and graph data [22, 25–28], and also in data mining [29–31].

Definition 1. A randomized function K gives ε-differential privacy if, for all data sets D_1 and D_2 differing on at most one element, and all $S \subseteq \text{Range}(K)$,

$$\Pr\left[K\left(D_1\right) \in S\right] \leq \exp\left(\varepsilon\right) * \Pr\left[K\left(D_2\right) \in S\right]. \quad (1)$$

Here, ε is a small positive value with which one can balance the tradeoff between privacy and accuracy. Relatively, if ε is smaller, the privacy is higher and accuracy is lower, and vice versa. Usually, ε is chosen by the user administering the privacy policy; therefore, selecting a reasonable ε is worth further study. Moreover, an algorithm that provides ε-differential privacy for neighboring databases differing on a single record also provides $k\varepsilon$-differential privacy [14] for neighboring databases differing on at most k records.

To achieve differential privacy, a certain amount of random noise must be added to the answer of the query set. Intuitively, its magnitude should cover up the largest change that a single record could have on the output.

Definition 2. Let Q be a sequence of counting queries. The sensitivity of Q is denoted by Δ_Q:

$$\Delta_Q = \max_{D_1, D_2} \left\| Q\left(D_1\right) - Q\left(D_2\right) \right\|_1. \quad (2)$$

In particular, a simple counting query has $\Delta_Q = 1$. For example, consider a private personnel database with an attribute column to indicate marital status. An analyst may query the number of married persons, x, and the number of unmarried persons, y, so this query set (x, y) has $\Delta_Q = 1$, because adding or removing one record changes exactly one output by a value of one. Furthermore, if he simultaneously queries the total number of people, z, the query set (x, y, z) has $\Delta_Q = 2$, because one change could affect two outputs, each by a value of one. Note that, in the second query set, there exist constraints, $z = x + y$, by which someone can search for the closest consistent solution to boost the accuracy of the results.

The Laplace mechanism [19], the most common noise-adding mechanism, disturbs the outputs by adding noise produced by a Laplace distribution to achieve differential privacy.

Proposition 3. *Let Q be a query sequence of length d. The randomized algorithm M, which takes database D as input and outputs the following vector, satisfies ε-differential privacy.*

$$M(D) = Q(D) + \left\langle \mathrm{Lap}\left(\frac{\Delta_Q}{\varepsilon}\right)\right\rangle^d. \tag{3}$$

Here, $\langle \mathrm{Lap}(\Delta_Q/\varepsilon)\rangle^d$ denotes a d-length vector of i.i.d. (independent and identically distributed) samples from a Laplace distribution with scale Δ_Q/ε. In other words, the magnitude of the noise is proportional to Δ_Q, and inversely proportional to ε. Proof of the proposition can be found in the literature.

Sometimes we need to combine several differentially private mechanisms in complex privacy issues as in this paper, so we can take advantage of the combination properties [21] of differential privacy.

Proposition 4. *Let each A_i provide ϵ_i-differential privacy. A sequence of $A_i(D)$ over the database D provides $\Sigma\epsilon_i$-differential privacy.*

It is the sequential theorem of differential privacy. Intuitively, ε can be split among a sequence of differentially private mechanisms and the final output still provides differential privacy.

2.2. Unattributed Histogram. A conventional histogram adopts a box-dividing technology, a popular form for data reduction, to approximate the data distribution. It divides ranged attributes into disjoint subsets or barrels, which usually are continuous intervals for a given attribute, and then computes counting queries for each specified range. Distinguishingly, in an unattributed histogram, each barrel only represents a single attribute value, that is, a unit-length range. An important instance is the degree distribution of networks, in which each barrel is the degree of one node, and the histogram is simply the sorted degree sequence.

In the context of our application, firstly consider a communication database D, organized as a set of records (s, t), in which a record represents one communication between two addresses. Whenever a communication occurs, a record is added to the database. Next, this database is converted to a multigraph. If the same record (s, t) appears W times, there are W edges between s and t in the graph. Finally, the multigraph is transformed into a weighted graph, in which the edge weight is the number of edges, W, between any two vertices. Therefore, we view each edge weight as one barrel, and the edge-weight sequence as an unattributed histogram. Naturally, differential privacy for edge weights can be implemented based on the histogram.

3. Methods

In this section, we detail the Lap strategy and the MB-CI (Merging Barrels and Consistency Inference) strategy, which are used to perform differential privacy for edge weights. Furthermore, the algorithm of MB-CI strategy is provided. To evaluate and quantify the error of the added noise, a formula is given using the common Squared Error to calculate the expectation of the possible randomness.

Definition 5. For a primitive edge-weight sequence S and its noisy sequence S^*, the introduced error (S^*) is $E(\sum_{i=1}^{n}(S_i^* - S_i)^2)$. Here, n is the length of the sequence.

3.1. Lap Strategy. To achieve differential privacy for edge weights, the naive strategy, called Lap strategy, is to directly add Laplace noise without any processing.

Theorem 6. *The edge-weight sequence S has $\Delta_Q = W_{max} - W_{min}$, W_{max} is the maximum of edge weights, and W_{min} is the minimum of edge weights.*

Proof. Given a graph G_1 and its neighbor graph G_2 differing on at most one edge weight, the edge-weight sequence S has only one value changed by at most $W_{\max} - W_{\min}$ and all other values kept the same. According to Definition 2, the edge-weight sequence S has $\Delta_Q = W_{\max} - W_{\min}$. For simplicity, $W_{\max} - W_{\min}$ is denoted by Δ_W.

On the basis of Proposition 3, the scale of Laplace noise added is Δ_W/ε, so each edge weight should add $\mathrm{Lap}(\Delta_W/\varepsilon)$. The error in this strategy can be computed as follows:

$$\begin{aligned}
\mathrm{Error}(S^*) &= E\left(\sum_{i=1}^{n}(S_i^* - S_i)^2\right) \\
&= E\left(n * \left(\mathrm{Lap}\left(\frac{\Delta_W}{\varepsilon}\right)\right)^2\right) \\
&= n * E\left(\mathrm{Lap}\left(\frac{\Delta_W}{\varepsilon}\right)\right)^2 \\
&= n * V\left(\mathrm{Lap}\left(\frac{\Delta_W}{\varepsilon}\right)\right) = \frac{2n}{\varepsilon^2}(\Delta_W)^2.
\end{aligned} \tag{4}$$

□

3.2. MB-CI Strategy. We propose a novel strategy which needs less noise to achieve differential privacy for edge weights; for the global utility, it keeps most of the shortest paths unchanged. The proposed MB-CI strategy includes two key steps, merging barrels and consistency inference.

3.2.1. Merging Barrels. Consider that in a social network, especially a large one, some edges or even more should have the same weight. Viewing the edge-weight sequence as an unattributed histogram, we merge barrels with the same count into one group to reduce the added noise. It does not introduce histogram construction error, as the value of each barrel does not change after merging. Then, we can add less Laplace noise to each barrel merged, while adding the same amount as the Lap strategy to other ones.

Theorem 7. *The noise added to every merged barrel is $\mathrm{Lap}(\Delta_W/(C * \varepsilon))$; C is the number of merged barrels.*

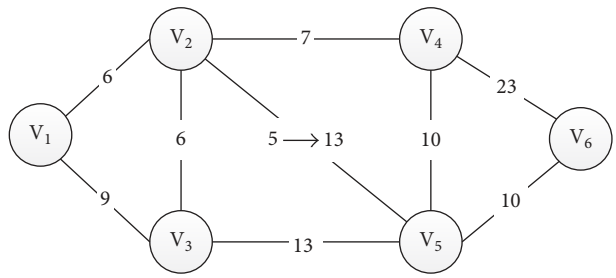

Figure 1: A simple weighted graph G.

Proof. Given a graph G_1 and its neighbor graph G_2 differing on at most one edge weight, which will affect at most one group by value Δ_W, and Δ_W/C for every barrel in the group, hence, the noise added to every merged barrel is $\mathrm{Lap}(\Delta_W/(C * \varepsilon))$.

Others unmerged will still need to add $\mathrm{Lap}(\Delta_W/\varepsilon)$ as the Lap strategy does. The error of this approach can be calculated as follows. Given n weights, merging into k groups, $k \le n$, the first group has n_1 values, and so forth, and the kth group has n_k values, and $n_1 + n_2 + \cdots + n_k = n$. Then,

$$\begin{aligned}
\mathrm{Error}(S^*) &= E\left(\sum_{i=1}^{n}(S_i^* - S_i)^2\right) = E\left(\sum_{i=1}^{n_1}(S_i^* - S_i)^2\right. \\
&\left. + \sum_{i=1}^{n_2}(S_i^* - S_i)^2 + \cdots + \sum_{i=1}^{n_m}(S_i^* - S_i)^2\right) = E\left(n_1\right. \\
&* \left(\mathrm{Lap}\left(\frac{\Delta_W}{n_1\varepsilon}\right)\right)^2 + n_2 * \left(\mathrm{Lap}\left(\frac{\Delta_W}{n_2\varepsilon}\right)\right)^2 + \cdots \\
&\left. + n_m * \left(\mathrm{Lap}\left(\frac{\Delta_W}{n_m\varepsilon}\right)\right)^2\right) = n_1 * E\left(\mathrm{Lap}\left(\frac{\Delta_W}{n_1\varepsilon}\right)\right)^2 \\
&+ n_2 * E\left(\mathrm{Lap}\left(\frac{\Delta_W}{n_2\varepsilon}\right)\right)^2 + \cdots + n_m \\
&* E\left(\mathrm{Lap}\left(\frac{\Delta_W}{n_m\varepsilon}\right)\right)^2 = n_1 * V\left(\mathrm{Lap}\left(\frac{\Delta_W}{n_1\varepsilon}\right)\right) + n_2 \\
&* V\left(\mathrm{Lap}\left(\frac{\Delta_W}{n_2\varepsilon}\right)\right) + \cdots + n_m * V\left(\mathrm{Lap}\left(\frac{\Delta_W}{n_m\varepsilon}\right)\right) \\
&= \left(\frac{2}{n_1\varepsilon^2} + \frac{2}{n_2\varepsilon^2} + \cdots + \frac{2}{n_m\varepsilon^2}\right)(\Delta_W)^2 = \frac{2}{\varepsilon^2}\left(\frac{1}{n_1}\right. \\
&\left. + \frac{1}{n_2} + \cdots + \frac{1}{n_m}\right)(\Delta_W)^2 \ll \frac{2m}{\varepsilon^2}(\Delta_W)^2 \\
&\le \frac{2n}{\varepsilon^2}(\Delta_W)^2.
\end{aligned} \tag{5}$$

□

Moreover, differential privacy may be violated if we simply merge all barrels with the same count into one group, because the magnitude of the noise itself may disclose some information. Therefore, k-indistinguishability between groups is proposed to guarantee that these groups require the same amount of noise. That is, these groups are indistinguishable only from the aspect of the amount of noise. In fact, we compromise in the merging step. We merge them while guaranteeing k-indistinguishability between groups; otherwise, we do nothing.

Definition 8. The groups are said to satisfy k-indistinguishability for an integer $k \ge 1$, if the number of groups with the same amount of barrels is greater than or equal to k.

For example, a simple weighted graph is shown in Figure 1, the range of weights is limited in 1~25. $W_{1,2} = W_{2,3} = 6$, $W_{4,5} = W_{5,6} = 10$, and the other weights are different. If we set $k = 2$, $W_{1,2}$ and $W_{2,3}$ can be merged into one group and $W_{4,5}$ and $W_{5,6}$ into the other group. Thus, there are two merged groups, and, with the noise added to $W_{1,2}$, $W_{2,3}$, and $W_{4,5}$, $W_{5,6}$ is $\mathrm{Lap}(12/\varepsilon)$. If we set $k = 3$, there are no merged groups and the noise added to each weight is $\mathrm{Lap}(24/\varepsilon)$. Suppose that $W_{2,5}$ is 13 instead of 5 in Figure 1; then $W_{2,5} = W_{3,5} = 13$. When $k = 2$ or 3, there will be three merged groups in both cases, and for larger k there will be no merged groups.

3.2.2. Consistency Inference. Here, we do consistency inference according to original order of the sequence as an important postprocessing step. The disturbed sequence should satisfy the original order to maintain consistency, which also means the relative weights between each edge do not change. Intuitively, the shortest paths will not go around easily but tend to be unchanged. It is worth mentioning that the process is only based on the known order, without accessing the private database; hence, there is no privacy leakage.

As a matter of fact, this problem is an instance of isotonic regression, and the following min-max formula [32] is one of the solutions.

Proposition 9. *Let $M[i, j]$ be the mean of elements from indexes i to j. Denote $L_k = \min_{j\in[k,n]}\max_{i\in[1,j]}M[i, j]$ and $U_k = \max_{i\in[1,k]}\min_{j\in[i,n]}M[i, j]$. The minimum L_2 solution is unique and given by $L_k = U_k$.*

In the literature [22], Hay et al. provided theoretical proof of the error brought by consistency inference; the results of derivation showed that it barely hurts the accuracy. However, the results of experiments showed that it could improve the accuracy obviously.

Input: Raw weighted-graph database D, privacy budget ε, parameter k
Output: Disturbed weighted-graph database D^*
// D and D^* contain three column vectors S, E, W, and S, E, W^*, respectively.
// S represents the starting points of edges. E indicates the ends.
// W and W^* store the original edge weights and the disturbed ones.
(1) Scan D once to compute three vectors C, K, O:
$C_i \leftarrow \text{Count}(W_i)$, $K_i \leftarrow \text{Count}(C_i)$, $O_i \leftarrow \text{Sort}(W_i)$.
(2) $\varepsilon = \varepsilon_1 + \varepsilon_2$
(3) $K^*_i = K_i + \text{Lap}(4/\varepsilon_1)$
(4) *for* $i = 1$ to N
(5) *if* $K^*_i >= k$ *then*
(6) $W^*_i = W_i + \text{Lap}(\Delta_W/(C_i * \varepsilon_2))$
(7) *else*
(8) $W^*_i = W_i + \text{Lap}(\Delta_W/\varepsilon_2)$
(9) *end if*
(10) *end for*
(11) *if* $\min(W^*) < 0$ *then* $W^* \leftarrow W^* - \min(W^*) + 1$
(12) *for* $i = 1$ to N
(13) $P^*_i = W^*(O_i)$
(14) *end for*
(15) *while* $i < N$
(16) $j = i + 1$
(17) *while* $j < N$
(18) *if* $M[i, j-1] < P^*_{i-1}$ or $M[i, j-1] > P^*_j$ *then*
(19) $j = j + 1$
(20) *else*
(21) $P^*_i \sim P^*_{j-1} \leftarrow M[i, j-1]$
(22) *end if*
(23) *end while*
(24) $i = j$
(25) *end while*
(26) *for* $i = 1$ to N
(27) $W^*(O_i) = P^*_i$
(28) *end for*
(29) *return* D^*

Algorithm 1: Merging Barrels and Consistency Inference (MB-CI) algorithm.

3.2.3. Algorithm of the MB-CI Strategy. Algorithm 1 is the complete algorithm of the MB-CI strategy.

Algorithm MB-CI presents entire process of the proposed strategy. Line 1 scans D once to compute three vectors, C, K, and O. Each element C_i of C stores the count of the corresponding W_i, which is also the number of barrels with the same count. Each element K_i of K stores the count of the corresponding C_i to estimate whether to merge or not. Each element O_i of O points to the index of the corresponding W_i in the original order. Line 2 allocates privacy budget ε according to the proportion of 2 : 8 in the experiments. That is, $\varepsilon_1 = 0.2 * \varepsilon$; $\varepsilon_2 = 0.8 * \varepsilon$. To randomly choose groups to merge, Line 3 adds Laplace noise to each K_i according to Theorem 10.

Theorem 10. *The vector K has $\Delta_Q = 4$.*

Proof. Given a graph G_1 and its neighbor graph G_2 differing on at most one edge weight, the vector W has only one value changed and all other values kept the same. The vector C, storing the count of the corresponding W_i, has two values changed with one plus 1 and the other minus 1. In the same way, the vector K, storing the count of the corresponding C_i, has four values changed. According to Definition 2, the vector K has $\Delta_Q = 4$. □

Lines 4–10 add Laplace noise to every weight; we need to test whether it satisfies k-indistinguishability between groups. If it is true, we merge the barrels, so the amount of noise should be $\text{Lap}(\Delta_W/(C_i * \varepsilon))$. Otherwise, $\text{Lap}(\Delta_W/\varepsilon)$ is still added, which is equivalent to not merging. Line 11 mainly deals with the negative weight, which is meaningless and inexistent. Specifically, if the minimum of W^* is less than zero, we uniformly adjust all the values to subtract the minimum, rather than simply resetting all the negatives to zero. The purpose is not to mandatorily change some weights, but to ensure that all the weights remain relatively unchanged. In addition, adding one to the results is to make the minimum nonzero; otherwise, it is likely to cancel an edge between two vertices.

Lines 12–14 generate a vector P^* that stores the noisy weights according to the corresponding indexes. Obviously,

Table 1: Data sets statistics.

Datasets	Type	Vertices	Edges	Weights range
BA1	Undirected	1,000	4,967	100~600
BA2	Undirected	2,000	9,966	100~800
CA-GrQc	Undirected	5,242	14,490	100~800

P^* should satisfy the original order. Lines 15–25 adopt nonrecursive programming based on the ideas of min-max formula to do consistency inference. If the current value does not meet the conditions—that is, it is smaller than the previous value, or bigger than the next—we continue to merge back and calculate the mean. Otherwise, the mean is assigned to each element in this cycle. It is worth mentioning that, considering some special situation, the last group may be out of order, so we do consistency inference from back to front to readjust it again in the experiments. Lines 26–28 reset the processed noisy weights and W^* is obtained in the end. Line 29 returns D^* as the output of the algorithm.

Theorem 11. *Algorithm (MB-CI) guarantees ε-differential privacy.*

Proof. In the algorithm, adding Laplace noise guarantees differential privacy according to Theorems 6 and 7. Furthermore, randomly choosing groups to merge guarantees differential privacy according to Theorem 10. The rest lines do not incur any extra privacy cost. Therefore, MB-CI algorithm as a whole guarantees ϵ-differential privacy according to Proposition 4. □

4. Experiments

In this section, the proposed approach is evaluated from two aspects, accuracy and utility. We use ARE (Average Relative Error) to test the loss of accuracy due to the added noise and KSP (Keeping Shortest Paths) to measure the proportion of unchanged shortest paths.

(1) WARE is the average relative error of all the edge weights. The smaller the value, the higher the accuracy.

$$\text{WARE} = \frac{\left(\sum_{i=1}^{N} \left|W_i^* - W_i\right|\right)}{N}. \tag{6}$$

(2) KSP is the proportion of unchanged shortest paths. N_P is the number of all the reachable shortest paths, and $N_{P'}$ is the number of all the unchanged shortest paths. The greater the value, the more the unchanged shortest paths and the better the utility.

$$\text{KSP} = \frac{N_{P'}}{N_P}. \tag{7}$$

(3) LARE is the average relative error of all the unchanged shortest paths, not considering the shortest paths that have changed, as it does not make sense to compare the lengths of different paths.

$$\text{LARE} = \frac{\left(\sum_{i=1}^{N_{P'}} \left|L_i^* - L_i\right|\right)}{N_{P'}}. \tag{8}$$

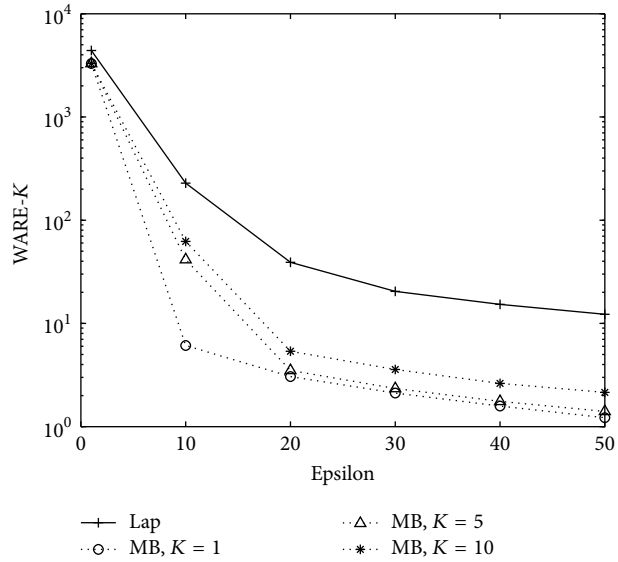

Figure 2: Results of WARE for BA1.

Three data sets were used in the experiments, shown in Table 1. There are two synthetic data sets employing a BA (Barabási–Albert) model to generate scale-free networks. The first one has five fully connected vertices in the original state. With each new vertex, five edges are associated at the same time, until it grows to 1,000 vertices. In the same way, we got the second one with a total of 2,000 vertices. The other is a real data set: CA-GrQc, the Collaboration Network of Arxiv General Relativity Category. There is an edge if two authors have coauthored at least one paper. We randomly assigned weights for each edge, ignoring its semantics. The experimental environment is an Intel® Core™ i7-6700 CPU @ 3.40 GHz, with 24 G memory, using the Windows 10 operating system; the algorithm offered was implemented in Matlab R2014a.

The MB-CI strategy is mainly composed of two steps: merging barrels, MB for short, and consistency inference, CI for short. To improve the accuracy, MB reduces the added noise by merging barrels with the same count while guaranteeing k-indistinguishability between groups. For the sake of guaranteeing better utility and keeping most of the

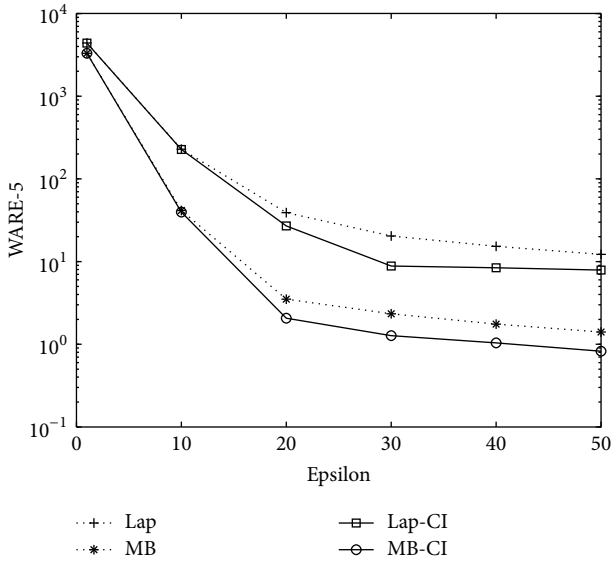

Figure 3: Results of WARE for BA1 when $k = 5$.

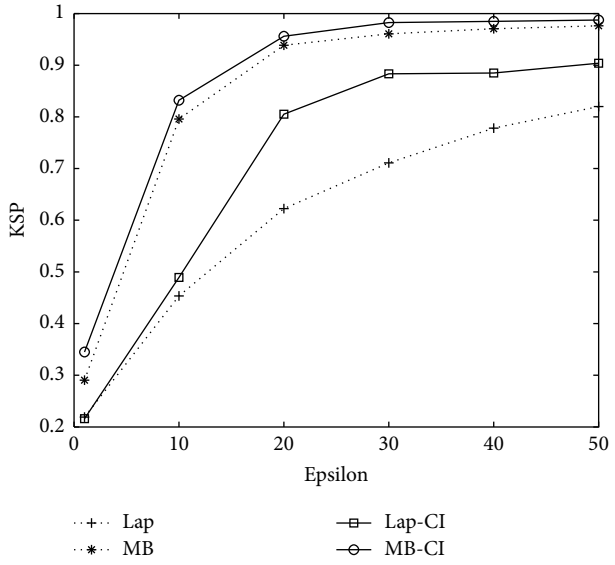

Figure 4: Results of KSP for BA1.

shortest paths unchanged, CI does not change the relative weights between each edge by consistency inference. In order to display the respective effects of the two steps, we break them up to compare with the Lap strategy. The MB strategy merges barrels with the same count while guaranteeing k-indistinguishability between groups and then adds Laplace noise to each barrel. The Lap-CI strategy does consistency inference based on the Lap strategy, which adds Laplace noise directly to each barrel with no merging. In the experiments, we set ε between 1 and 50, relatively large values, to balance the tradeoff between privacy and data utility, as we set the weights to a relatively large range. The fact is that large ε, more than 10, provide almost no privacy protection in practice and it is to check the performance of the algorithm in real social networks.

In the experiments, we evaluated the error for MB under different k, comparing with Lap first. The results are shown in Figures 2, 6, and 10; k takes three different values 1, 5, and 10 uniformly. The error decreases with the increase of ε, due to less noise. The error for Lap is maximal because there is no merging. When $k = 1$, it means that all the barrels with the same count are merged unconditionally, so the number of merging barrels is the greatest and the error is minimum. If k is larger, it means the limiting condition is stricter; there may be more barrels with the same count that could not be merged. Thus, when k takes two other values, the error is between them. The curves are not smooth as shown in the figures, because it depends on the proportion of the nonmerging barrels. Next, we set k to be value 5 for MB, to test how much error was introduced by consistency inference. The results are shown in Figures 3, 7, and 11. It can be seen that compared with Lap, Lap-CI effectively reduces the error; MB-CI adds some error compared with MB in the last data set. MB-CI may introduce extra error because it needs to process more data according to consistency inference in the larger data set, and the error in MB is already very small.

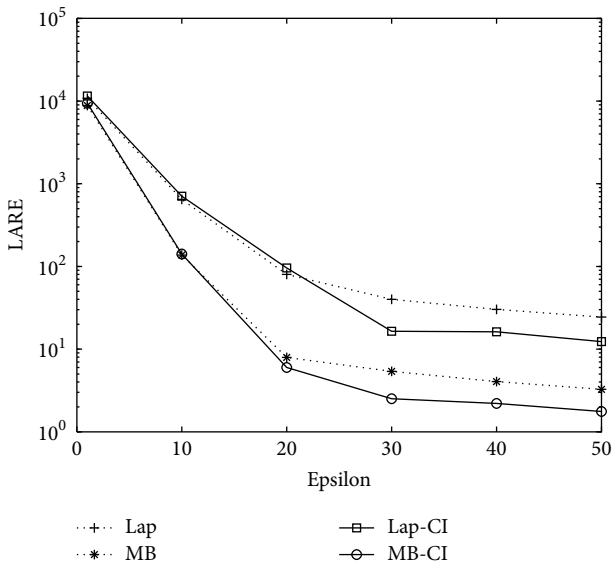

Figure 5: Results of LARE for BA1.

The most important step is to evaluate the change of the shortest paths, which is a key measure of the global utility. As shown in Figures 4, 8, and 12, with the increase of ε, more shortest paths will remain unchanged. Obviously, compared with Lap, MB can better protect the shortest paths. MB-CI has a little bit better effect than MB, as MB has kept about 90% of the shortest paths unchanged when ε is more than 20. Lap-CI has a much better effect than Lap when ε is more than 20. As shown in Figures 5, 9, and 13, we evaluated the error of all the unchanged shortest paths. It can be seen that the trends of these curves are consistent with the previous analysis. This suggests that consistency inference can further improve the

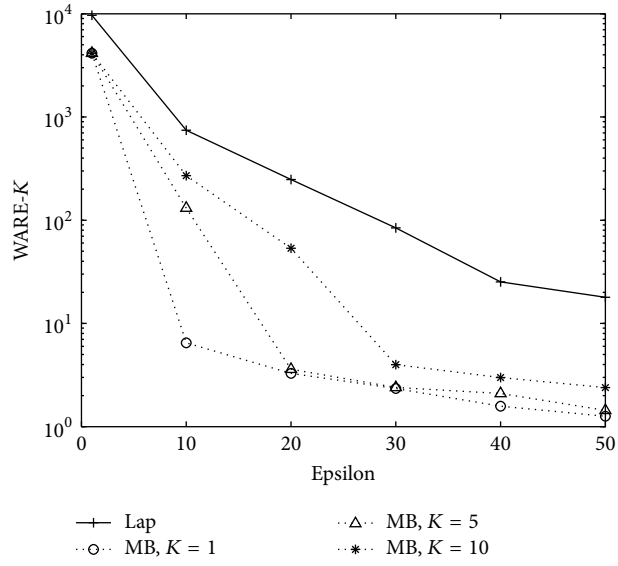

Figure 6: Results of WARE for BA2.

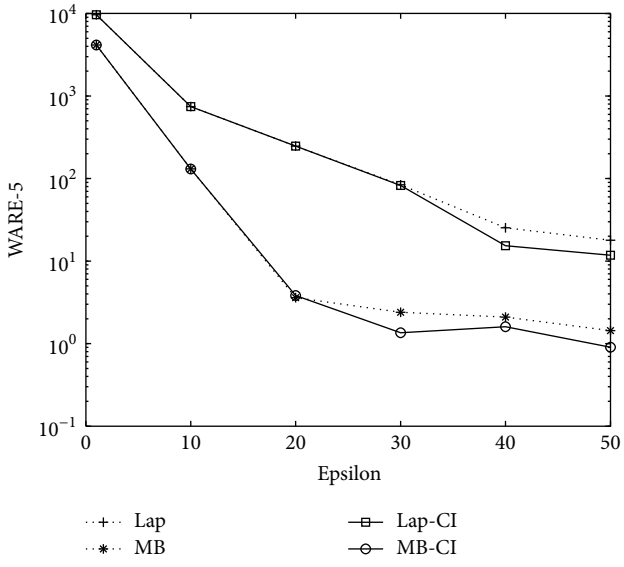

Figure 7: Results of WARE for BA2 when $k = 5$.

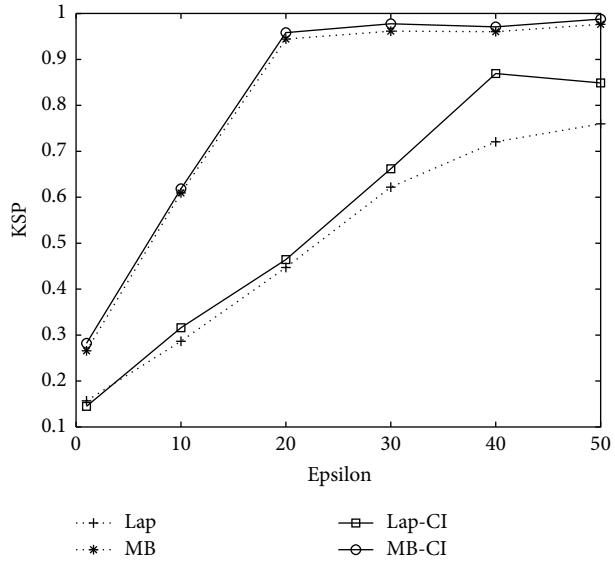

Figure 8: Results of KSP for BA2.

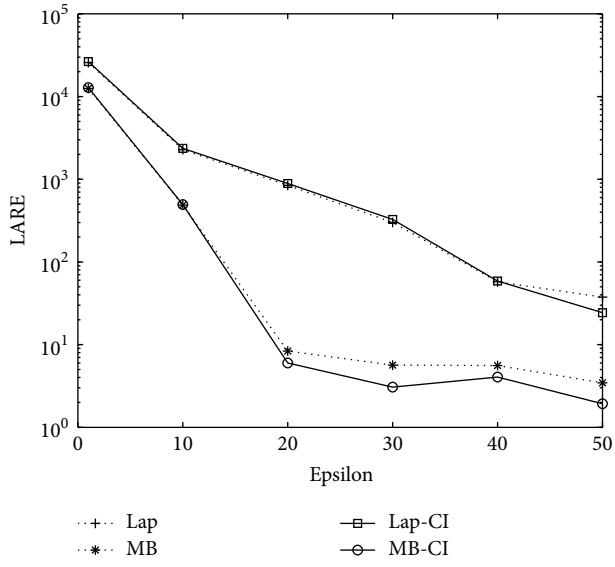

Figure 9: Results of LARE for BA2.

proportion of unchanged shortest paths and reduce the error effectively; this is an essential step in our application. In conclusion, MB-CI has achieved better performance of the experimental results.

5. Conclusions

In this paper, we proposed the MB-CI strategy, a novel approach for protecting the edge weights of social networks. The starting point was treating the edge-weight sequence as an unattributed histogram; we merged all barrels with the same count into one group, while guaranteeing k-indistinguishability between groups. Then, we added Laplace noise to every edge weight and did consistency inference according to original order of the sequence. We conducted experiments on both synthetic data sets and a real data set. The results showed that the MB-CI strategy improved the accuracy and utility of the released data, which are consistent with the theoretical analysis. That is, the approach was effective in reducing the error introduced by the added noise, and kept most of the shortest paths unchanged.

Note that, the edge weights considered here, are integers not continuous values. Thus, generalizing the data set to the real-value field is an object for future study. Moreover, many applications in the real world demand higher user-level privacy rather than record-level privacy. Therefore, we will further extend the method for providing stronger protection.

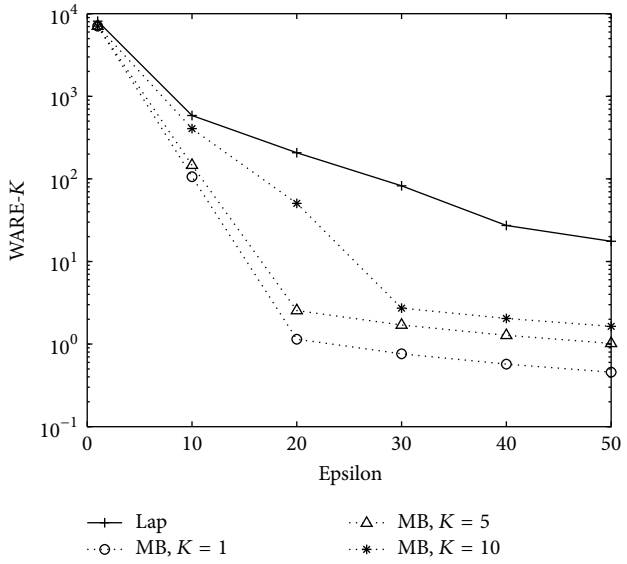

Figure 10: Results of WARE for CA-GrQc.

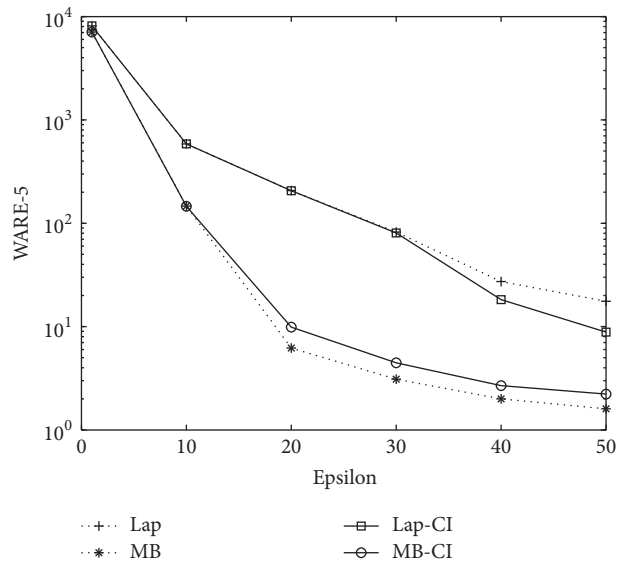

Figure 11: Results of WARE for CA-GrQc when $k = 5$.

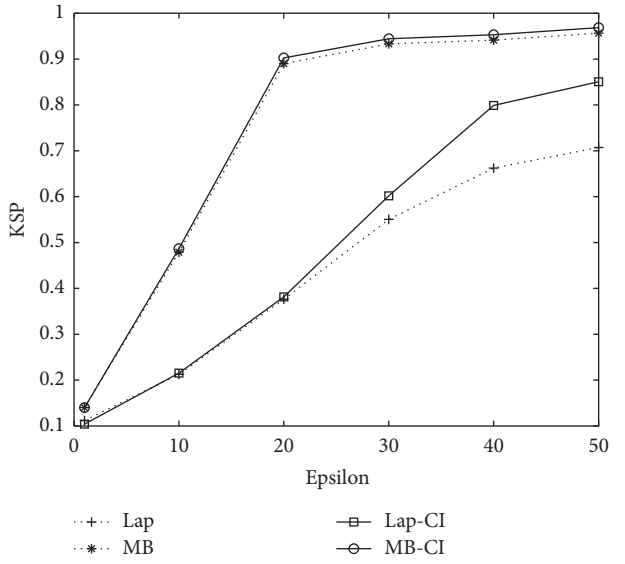

Figure 12: Results of KSP for CA-GrQc.

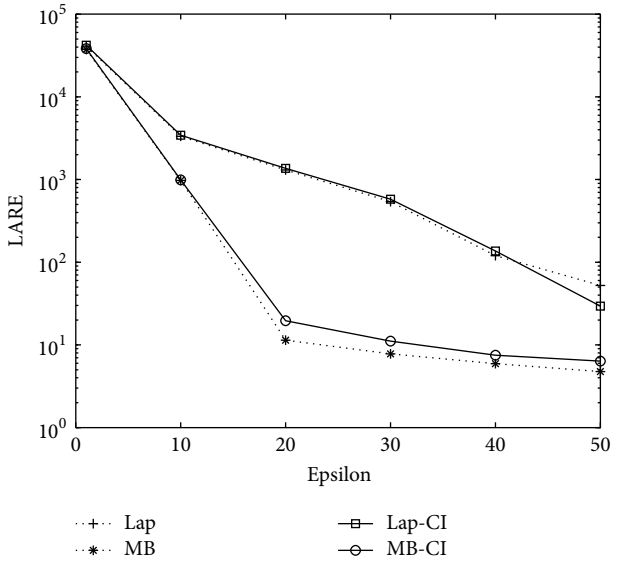

Figure 13: Results of LARE for CA-GrQc.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by The Natural Science Foundation of China (no. 61370083, no. 61402126, and no. 61672179), Specialized Research Fund for the Doctoral Program (no. 20122304110012), Youth Science Fund of Heilongjiang Province (no. QC2016083), Postdoctoral Fellowship of Heilongjiang Province (no. LBH-Z14071), and Basic Research Business of Education Department of Heilongjiang Province (no. 135109314, no. 135109245).

References

[1] T.-S. Hsu, C.-J. Liau, and D.-W. Wang, "A logical framework for privacy-preserving social network publication," *Journal of Applied Logic*, vol. 12, no. 2, pp. 151–174, 2014.

[2] A. R. Kulkarni and H. K. Yogish, "Advanced unsupervised anonymization technique in social networks for privacy preservation," *International Journal of Science and Research*, vol. 3, no. 4, pp. 118–125, 2014.

[3] B. K. Tripathy, M. S. Sishodia, S. Jain, and A. Mitra, "Privacy and anonymization in social networks," *Intelligent Systems Reference Library*, vol. 65, pp. 243–270, 2014.

[4] S. Das, Ö. Egecioglu, and A. El Abbadi, "Anonymizing weighted social network graphs," in *Proceedings of the 26th IEEE International Conference on Data Engineering (ICDE '10)*, pp. 904–907, Long Beach, Calif, USA, March 2010.

[5] L. Liu, J. Wang, J. Liu, and J. Zhang, "Privacy preserving in social networks against sensitive edge disclosure," Tech. Rep. CMIDA-HiPSCCS 006-08, Department of Computer Science, University of Kentucky, 2008.

[6] S. Costea, M. Barbu, and R. Rughinis, "Qualitative analysis of differential privacy applied over graph structures," in *Proceedings of the 11th Roedunet International Conference on Networking in Education and Research (RoEduNet '13)*, January 2013.

[7] M. Hayy, V. Rastogiz, G. Miklauy, and D. Suciu, "Boosting the accuracy of differentially private histograms through consistency," *Proceedings of the VLDB Endowment*, vol. 3, no. 1-2, pp. 1021–1032, 2010.

[8] J. Xu, Z. Zhang, X. Xiao, Y. Yang, and G. Yu, "Differentially private histogram publication," in *Proceedings of the IEEE 28th International Conference on Data Engineering (ICDE '12)*, pp. 32–43, IEEE, Washington, DC, USA, April 2012.

[9] J. Xu, Z. Zhang, X. Xiao, Y. Yang, G. Yu, and M. Winslett, "Differentially private histogram publication," *The VLDB Journal*, vol. 22, no. 6, pp. 797–822, 2013.

[10] J. Soria-Comas, J. Domingo-Ferrer, D. Sánchez, and S. Martínez, "Enhancing data utility in differential privacy via microaggregation-based *k*-anonymity," *The VLDB Journal*, vol. 23, no. 5, pp. 771–794, 2014.

[11] D. Sánchez, J. Domingo-Ferrer, S. Martínez, and J. Soria-Comas, "Utility-preserving differentially private data releases via individual ranking microaggregation," *Information Fusion*, vol. 30, pp. 1–14, 2016.

[12] T. Dalenius, "Towards a methodology for statistical disclosure control," *Statistik Tidskrift*, vol. 15, pp. 429–422, 1977.

[13] C. Dwork, "Differential privacy," in *Proceedings of the 33rd International Colloquium on Automata Languages and Programming (ICALP '06)*, pp. 1–12, Venice, Italy, July 2006.

[14] C. Dwork, "Differential privacy: a survey of results," in *Proceedings of the 5th International Conference on Theory and Applications of Models of Computation (TAMC '08)*, pp. 1–19, Xi'an, China, April 2008.

[15] C. Dwork, "The differential privacy frontier (extended abstract)," in *Proceedings of the 6th Theory of Cryptography Conference (TCC '09)*, pp. 496–502, San Francisco, Calif, USA, March 2009.

[16] C. Dwork and J. Lei, "Differential privacy and robust statistics," in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, pp. 371–380, Bethesda, Md, USA, May 2009.

[17] C. Dwork, "Differential privacy in new settings," in *Proceedings of the Symposium on Discrete Algorithms (SODA '10)*, Society for Industrial and Applied Mathematics, Austin, Tex, USA, January 2010.

[18] C. Dwork, "The promise of differential privacy: a tutorial on algorithmic techniques," in *Proceedings of the IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS '11)*, October 2011.

[19] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proceedings of the 3rd Theory of Cryptography Conference*, pp. 265–284, New York, NY, USA, March 2006.

[20] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *Proceedings of the 48th Annual Symposium on Foundations of Computer Science (FOCS '07)*, pp. 94–103, Providence, RI, USA, October 2007.

[21] F. McSherry, "Privacy integrated queries: an extensible platform for privacy-preserving data analysis," *Communications of the ACM*, vol. 53, no. 9, pp. 89–97, 2010.

[22] M. Hay, C. Li, G. Miklau, and D. Jensen, "Accurate estimation of the degree distribution of private networks," in *Proceedings of the 9th IEEE International Conference on Data Mining (ICDM '09)*, pp. 169–178, Miami, Fla, USA, December 2009.

[23] N. Li, W. Yang, and W. Qardaji, "Differentially private grids for geospatial data," in *Proceedings of the IEEE 29th International Conference on Data Engineering (ICDE '13)*, pp. 757–768, IEEE, Brisbane, Australia, April 2013.

[24] X. Xiao, G. Wang, and J. Gehrke, "Differential privacy via wavelet transforms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 8, pp. 1200–1214, 2011.

[25] A. Sala, X. Zhao, C. Wilson, H. Zheng, and B. Y. Zhao, "Sharing graphs using differentially private graph models," in *Proceedings of the ACM SIGCOMM Internet Measurement Conference (IMC '11)*, pp. 81–98, Berlin, Germany, November 2011.

[26] A. Gupta, A. Roth, and J. Ullman, "Iterative constructions and private data release," in *Proceedings of the 9th Theory of Cryptography Conference*, pp. 339–356, Taormina, Italy, March 2012.

[27] C. Task and C. Clifton, "A guide to differential privacy theory in social network analysis," in *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 411–417, Istanbul, Turkey, August 2012.

[28] R. Chen, B. C. M. Fung, P. S. Yu, and B. C. Desai, "Correlated network data publication via differential privacy," *VLDB Journal*, vol. 23, no. 4, pp. 653–676, 2014.

[29] N. Li, W. Qardaji, D. Su et al., "PrivBasis: frequent itemset mining with differential privacy," in *Proceedings of the 38th International Conference on Very Large Data Bases*, pp. 1340–1351, Istanbul, Turkey, August 2012.

[30] G. Jagannathan, K. Pillaipakkamnatt, and R. N. Wright, "A practical differentially private random decision tree classifier," in *Proceedings of the IEEE International Conference on Data Mining Workshops (ICDMW '09)*, pp. 114–121, December 2009.

[31] J. Zhang, Z. Zhang, X. Xiao et al., "Functional mechanism: regression analysis under differential privacy," in *Proceedings of the 38th Conference of Very Large Database*, pp. 1364–1375, Istanbul, Turkey, 2012.

[32] R. E. Barlow and H. D. Brunk, "The isotonic regression problem and its dual," *Journal of the American Statistical Association*, vol. 67, no. 337, pp. 140–147, 1972.