Graph Node Strength Histogram Publication Method with Node Differential Privacy

Wenfen Liu¹, Bixia Liu*, Qiang Xu¹, Hui Lei²

¹Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology, Guilin, Guangxi 541004, China
²Shanghai Soft China Information System Consulting Co., Ltd, Shanghai 200000, China

¹009598059@qq.com

Abstract. The node differential privacy (node-DP) can be used to protect the private information of nodes and edges in the graph. In this paper, we propose an algorithm of publishing node strength histogram under node-DP, which could improve the accuracy of publishing the node strength histogram. In this algorithm, we use the Sequence Edge removal to reduce the sensitivity of query function and restrict the weight of edges to make the distribution of node strength denser. Furthermore, we use the histogram grouping algorithm Hierarchical Cluster Grouping to group the buckets to merge buckets with close values into one group. The experiments show that our algorithm maintains higher data utility than those traditional histogram publishing algorithms under the same privacy budget.

Keywords: Differential Privacy, Graph Data, Node Strength, Histogram Publishing.

1. Introduction
With the rapid development of the Internet, graphs have become more and more common in daily life. For example, transportation networks, social networks, communication networks, etc., can be represented by graph. Analysis and mining of graph data can obtain a lot of useful information. However, the graph data also contains many personal sensitive information. Even if the sensitive information is anonymized, the published graph data still has the risk of leaking personal sensitive information. In recent years, the differential privacy (DP) mechanism [1] is considered to be one of the most suitable processing mechanisms in the field of privacy protection. It can ensure that the change of any record in a dataset will not have a significant impact on the output result [2]. Regarding the privacy protection of graph data under DP, Hay et al. [3] proposed edge-DP and node-DP. Edge-DP guarantees the neighboring graphs differ by only one edge, and neighboring graphs in node-DP differ by one node and all its edges. Therefore, the privacy protection strength provided by node-DP is higher than that of edge-DP.

In recent years, many researchers have focused on publishing degree distribution based on DP [3-5], but there are few studies on privacy protection of node strength. However, the degree distribution of graphs does not fully represent the importance of nodes. For example, in a network of email exchanges, the strength of a node can indicate the intimacy of two people, but the degree of node cannot. Qian et
al. [6] studied the node strength histogram publishing mechanism based on edge-DP, and proposed the sequence-based and density-based clustering methods to reduce the approximation error generated by histogram grouping. However, node-DP can provide stronger privacy protection, because it ensures that a node and all its edges are not recognized. Therefore, the implementation of publishing node strength under node-DP should also be concerned.

This paper mainly studies the method of publishing the node strength histogram under node-DP, which faces the problem of high sensitivity that also exists in the publication of node degree distribution. At present, some scholars have studied this problem [4, 5, 7, 8]. The main method to solve this problem is to project the original graph onto a compressed graph whose node degree does not exceed a given threshold $\theta$. Meanwhile, actual node strength distribution is very sparse, we restrict the weight of edges to process the original graph, which puts a threshold $t$ on the weight of edges. In addition, this paper will use a bottom-up hierarchical clustering algorithm to group the buckets with close values into one group as much as possible, and replace the value of each bucket in the group with average value. Finally, we propose the $(\theta, t)$-Grouping-Hist algorithm to select suitable parameters through exponential mechanism. The main contributions of this paper are as follows:

1) We reduce the global sensitivity of the query function and the accumulative noise of long-range queries by preprocessing.
2) To reduce the error caused by grouping, we use the grouping algorithm Hierarchical Cluster Grouping (HCG) to group the buckets with close values in histogram into one group.
3) We propose the $(\theta, t)$-Grouping-Hist algorithm, which selects the grouping strategy that minimizes the error of the published result through exponential mechanism. We have also proved that the $(\theta, t)$-Grouping-Hist algorithm not only ensure the privacy of data, but also improves the data utility.

2. Related work
With the advent of the era of big data, k-anonymity, etc., traditional privacy protection technologies have long been unable to meet people’s requirements for privacy protection. DP mechanism was originally proposed for privacy disclosure in datasets. It doesn’t need to consider any possible background knowledge possessed by the attacker and is supported by a solid mathematical foundation, so that we can compare privacy protection level provided by one privacy protection technology under different parameters.

A series of studies in recent years have also made DP mature in theory and technology applications. Chatzikokolakis et al. [9] extended the Hamming distance of neighboring datasets defined in DP to an arbitrary metric, and applied this new framework of DP to protect important infrastructure networks; In [10], André et al. used differential privacy for the protection of geographic location, so that users would not leak real location information to the positioning program within a specified radius. At the same time, DP has been widely used in graph data protection research. In [11], applied DP to the study of graph data for the first time, and applied edge-DP to the triangle count of social networks. In [5], Zhang et al. published the degree histogram of the graph based on node-DP. In recent years, there have been many researches on histogram publishing technology based on DP [12-15], in which the classic one is the NoiseFirst method [13], which is based on the idea of grouping and adds noise to the merged groups to reduce the overall noise. Since the NoiseFirst method can only solve the problem of one-dimensional histogram publishing, the literature [12] combined with the idea of kd tree and partition, proposed DPCube method to make up for the deficiency of NoiseFirst method.

3. Preliminaries

3.1. Differential privacy
Dwork [1] proposed a differential privacy model in 2006. Let \( D \) be a cluster consisting of all datasets. Two datasets \( D, D' \notin D \) are defined as neighboring datasets if they only differ on one record, denoted by \( D \neq D' \).

**Definition 1 (\( \varepsilon \)-Differential Privacy).** A randomized algorithm \( K \) is the \( \varepsilon \)-differential privacy if for any \( S \subseteq \text{Range}(K) \), and any two neighboring datasets \( D \) and \( D' \), it satisfies:
\[
\Pr[K(D) \in S] \leq e^\varepsilon \times \Pr[K(D') \in S]
\]
where \( \varepsilon \) is called privacy budget.

When the nose is too large, the query results may lose the original value, so the researchers proposed the concept of global sensitivity, which is used to control the amount of noise.

**Definition 2 (Global Sensitivity).** For any two real-value query function \( f \) and two neighboring datasets \( D \) and \( D' \), the global sensitivity is defined as:
\[
\Delta_f = \max_{D, D'} \| f(D) - f(D') \|_1
\]
where \( \| f(D) - f(D') \|_1 \) is the L1 distance.

**Laplace Mechanism** For any real-value query function \( f \), the mechanism
\[
K(D) = f(D) + \sum_{i} Y_i
\]
satisfies \( \varepsilon \)-DP, where \( Y_i \) is the random noise, which follows the Laplace distribution with the scale parameter \( \Delta_f / \varepsilon \).

**Exponential Mechanism** Given a utility function \( q(D, r) \) for outputting \( r \) on input \( D \), and
\[
\Delta_q = \max_{D, D'} \| q(D, r) - q(D', r) \|_1
\]
is the sensitivity of \( q \). A randomized algorithm \( K \) satisfies \( \varepsilon \)-DP when its outputs \( r \) with a probability proportional to
\[
\exp(\varepsilon q(D, r) / 2\Delta_q)
\]

**Sequential Composition** Let \( K_1, \ldots, K_n \) be \( n \) algorithms that apply to the same dataset \( D \), and satisfy \( \varepsilon_1 - DP, \varepsilon_2 - DP, \ldots, \varepsilon_n - DP \) respectively, then publishing the results of all of \( K_1(D), \ldots, K_n(D) \) satisfies \( \sum_{i=1}^{n} \varepsilon_i - DP \).

**Parallel Composition** Let \( K_1, \ldots, K_n \) be \( n \) algorithms that apply to the disjoint dataset \( D_1, \ldots, D_n \), and satisfy \( \varepsilon_1 - DP, \ldots, \varepsilon_n - DP \) respectively, then publishing the results of all of \( K_1(D_1), \ldots, K_n(D_n) \) satisfies \( \max(\varepsilon_1) - DP \).

**Definition 3 (Neighboring graphs).** Any two graphs \( G = (V, E, W) \) and \( G' = (V', E', W') \) are neighboring graphs, if and only if \( V = V' \cup \{i\}, E = E' \cup \hat{E}, \) where \( \hat{E} \) is a set of edges of \( \hat{V} \).

4. **Proposed algorithms**

We present the preprocessing algorithm and provide the global sensitivity of publishing the node strength histogram after preprocessing in section 4.1. Then in sections 4.2 and 4.3, we show the histogram grouping algorithm and an algorithm for publishing node strength histogram. Finally, we analyze the privacy of our proposed algorithm in section 4.4.

The meanings of important variables and symbols mentioned in algorithms are as follows:

| variables and symbols | meaning |
|----------------------|---------|
| \( G \)             | weighted graph |
| \( V \)             | the set of nodes |
| \( E \)             | the set of edges |
| \( W \)             | the set of weights |
| Symbol | Meaning |
|-------|---------|
| $n$ | the number of nodes |
| $\theta$ | the threshold of node degree |
| $\Theta$ | the candidate set of degree thresholds |
| $k$ | number of histogram groups |
| $K$ | the candidate set of grouping thresholds |
| $t$ | threshold of weight |
| $T$ | the candidate set of weight thresholds |
| $s(v_i)$ | the strength of node $v_i$ |
| $\text{deg}(v_i)$ | the degree of node $v_i$ |
| $h_i$ | the $i$th bucket |
| $g_i$ | a group consisting of several buckets |
| $\Omega$ | a grouping strategy consisting of several groups |
| $\Phi$ | the set of several grouping strategies |
| $q$ | utility function |
| $Q$ | the set of the utility function |

**Tab. 1** The meaning of variables and symbols

### 4.1. Pretreatment

In this paper, we study the method of publishing node strength histogram in a weighted graph based on node-DP. Node-DP can provide higher privacy protection, but the global sensitivity of node strength histogram is also high, which will cause too much Laplace noise and affect the utility of results. In order to reduce the global sensitivity, Zhang et al. [5] proposed the edge removal algorithm Sequence Edge-removal (SER). SER is used to compress the original graph. Through the processing of the SER, we will get the compressed graph $G'_e$.

Roughly speaking, SER algorithm first calculates the degrees of all nodes in the graph $G_e$ and sorts them from small to large; then find the node $v_i$ with the largest degree, get the sequence of its neighboring nodes in ascending order of degree, and delete the edges connected to node $v_i$ in this sequence until $\text{deg}(v_i) = \theta$; reorder all nodes from small to large according to degree, repeat the above operation, when the degree of all nodes is not greater than $\theta$, the algorithm ends. The main purpose of SER algorithm is to reduce the sensitivity of query function. Compared with other projection algorithms [4, 8], the SER algorithm maximizes the original edges in $G_e$ and provides the basis for the application of DP mechanism in compressed graphs. We found that changing the degree of the node will change the node strength accordingly, which means that the global sensitivity of publishing the node strength histogram is the same as the global sensitivity of publishing the node degree histogram. Therefore, in this paper, we apply SER algorithm to reduce the global sensitivity.

According to Theorem 3 in [5], we know that the global sensitivity $\Delta_{\text{hist}(G'_e)}$ of publishing the node strength histogram after preprocessing by the SER algorithm is $2\theta + 1$.

Furthermore, actual node strength follows power-law distribution, so the strength of most nodes in the node strength histogram is concentrated in the first half. Especially when the maximum node strength is much greater than the number of nodes, there are many empty buckets in the histogram, which means that the accuracy of the range counting query will be reduce by accumulative noise. To solve this problem, we will traverse all edges in the graph $G'_e$ to ensure the weight on each edge does not exceed the threshold $t$ and get the graph $G_{(t,\theta)}$. In particular, a larger threshold $t$ can remain...
more weight information, and the number of buckets will also increase. On the contrary, a smaller threshold means that we will get a dense node strength distribution, but also lose more weight information.

4.2. Hierarchical cluster grouping algorithm

The histogram grouping algorithm groups buckets with close values into a same group, and then assigns the mean of its group sum to each bucket in the group, so as to reduce the cumulative noise. But most grouping algorithms ignore the approximation error caused by grouping. The literature [6] proposed a density-based histogram grouping algorithm, which uses a greedy strategy to find cluster centres at once without iteratively updating cluster centres. Although the histogram in this algorithm can be grouped quickly, it only guarantees the local optimal grouping of the histogram, which usually results in a large approximation error.

In this paper, we used the Hierarchical Cluster Grouping (HCG) algorithm to solve this problem. HCG algorithm first treats each bucket as an initial class, then merges two buckets according to the average distance every time until the number of classes equals \( k \). The specific description is as follows:

Algorithm 1. The input graph \( G(\theta,j) \) is the graph obtained after the pretreatment of graph \( G \) and \( H(\theta,j) \) is the node strength histogram sequence of \( G(\theta,j) \). Each \( g_i \) is a group of \( h_j \) in \( H(\theta,j) \), \( \Omega(\theta,j) = \{g_1, g_2, \cdots, g_k\} \).

Algorithm 1. Hierarchical Cluster Grouping Algorithm.

**Input:** Node strength histogram sequence \( H(\theta,j) = \{h_1, h_2, \cdots, h_n\} \) of graph \( G(\theta,j) \) and the number of buckets \( k \).

**Output:** a grouping strategy \( \Omega(\theta,j) = \{g_1, g_2, \cdots, g_k\} \).

1. treat each bucket \( h_j \) in the histogram sequence \( H(\theta,j) = \{h_1, h_2, \cdots, h_n\} \) as an initial cluster, and store it in set \( \Omega(\theta,j) \);
2. if the number \( r \) of clusters in the current set \( \Omega(\theta,j) \) is greater than \( k \), then calculate the distance between the clusters according to the average distance formula (1), and merge the two clusters \( g_a \) and \( g_b \) with the smallest distance in \( \Omega(\theta,j) \) to obtain a new cluster \( g^* \), then delete \( g_a \) and \( g_b \) from \( \Omega(\theta,j) \);
3. update \( \Omega(\theta,j) \) as: \( \Omega(\theta,j) = \Omega(\theta,j) \cup \{g^*\}, \quad r = r - 1 \);
4. repeat steps 2, 3 until \( r = k \) the algorithm ends.

In Algorithm 1, The average distance is defined as:

\[
dis(g_i, g_j) = \frac{\sum_{k \in g_i, k \in g_j} \sqrt{(h_k - h_j)^2}}{|g_i|^{1/2}|g_j|^{1/2}}
\]

(1)

The average distance determines the similarity between \( g_i \) and \( g_j \). A smaller average distance means that \( g_i \) and \( g_j \) are more similar. Algorithm 1 groups two buckets with close values each time, this grouping method not only solves the problem of accumulative noise but also solves the problem of local grouping. Compared with the density-based histogram grouping algorithm, which only merges adjacent buckets, the HCG algorithm merges similar buckets in the entire histogram. For example, the density-based histogram grouping algorithm will set \( g_1 = \{20, 15, 15\} \), \( g_2 = \{35, 30\} \),
\( g_1 = \{16.19\} \) when node strength histogram \( H_{(\theta,t)} = \{20, 15, 15, 35, 30, 16.19\} \) and \( k = 3 \), while HCG algorithm will take \( g_1 = \{20, 19\}, g_2 = \{15, 15, 16\}, g_3 = \{35, 30\} \), because HCG can not only merge adjacent buckets. In order to intuitively explain the advantage of the HCG algorithm, we will show this example in Figure 1. Meanwhile, Figure 1 compares the L1 error of those two grouping algorithms, the smaller L1 error means that the smaller the approximate error caused by the grouping. In Figure 1, the example shows that the L1 error of Algorithm 1 is smaller than that of the density-based histogram grouping algorithm.

4.3. Parameter selection

In section 4.1 and 4.2, the parameter \( \theta, t \) and \( k \) depend on the distribution of the graph data. A larger parameter can remain more graph information, but a smaller parameter means that we will make the lower sensitive and get a dense node strength distribution. In this section, we propose the \((\theta, t)\) -Grouping-Hist algorithm to balance the errors caused by processing, grouping and adding Laplace noise. In the \((\theta, t)\) -Grouping-Hist algorithm, the set of parameter candidate \( \Theta \), \( T \) and \( K \) can be split into several pairs of parameters \((\theta, t, k)\), and we will get different grouping strategies \( \Omega_{(\theta,t)} \) by different parameters \((\theta, t, k)\) pairs. In order to select the strategy \( \Omega_{(\theta', t')} \) that minimizes the error of histogram publication under differential privacy, \((\theta, t)\) -Grouping-Hist algorithm calculates the utility function corresponding to each grouping strategy \( \Omega_{(\theta,t)} \) firstly, then use exponential mechanism to select the grouping strategy \( \Omega_{(\theta', t')} \). According to the grouping strategy \( \Omega_{(\theta', t')} \), we will get a histogram \( H_{(\theta', t')} \). Last, adding Laplace noise to histogram \( H_{(\theta', t')} \) and get published histogram.

**Algorithm 2.** \((\theta, t)\) -Grouping-Hist algorithm.

**Input:** Weighted graph \( G = (V, E, W) \), privacy budget \( \varepsilon_1, \varepsilon_2 \), candidates \( \Theta, T, K \).

**Output:** Histogram \( H_{\text{node}} \) of node strength.

1. preprocessing and grouping the original graph \( G \) according to the parameter pair \((\theta, t, k)\), then the grouping strategy \( \Omega_{(\theta,t)} \) obtained by each pair of parameters is stored in \( \Phi \);
2. compute the utility function \( q(G, \Omega_{(\theta,t)}, e_i) \) corresponding to each pair of parameters \((\theta, t, \Omega_{(\theta,t)})\), and store \((\theta, t, \Omega_{(\theta,t)})\) in \(Q\);
3. select the grouping strategy \( \Omega^*_{(\theta,t')} \) from \( \Phi \) with a probability proportional to \( \exp \left( \frac{\varepsilon_i q(G, \Omega_{(\theta,t)}, e_i)}{2\Delta_q} \right) \);
4. according to the parameters \((\theta', t', \Omega^*_{(\theta,t')})\), the algorithms 1, 2, and 3 are used in sequence on the graph \( G \), then obtains a histogram \( H_{(\theta,t')} \);
5. add Laplace noise to \( H_{(\theta,t')} \):
\[
H'_{\text{noise}} = H_{(\theta,t')} + \text{lap} \left( \frac{\Delta_{\text{mot}(G,e_i)}}{\varepsilon_i} \right) \hat{h}_{i'_{(\theta,t')}}
\]
6. sort \( H'_{\text{noise}} \) according to the subscript \( i \) of \( h_i \) from small to large to obtain the strength histogram \( H_{\text{noise}} \), and the algorithm ends.

In this paper, the utility function \( q(G, \Omega_{(\theta,t)}, e_i) \) in exponential mechanism consists of the four parts:
\[
q(G, \Omega_{(\theta,t)}, e_i) = -\left( e_p + e_o + e_{\text{group}} + e_{\text{noise}} \right)
\]  
(2)

First error \( e_p \) is caused by SER. \( e_p \) is obtained by calculating the number of nodes whose degree is greater than \( \theta \) [4].

The second error \( e_o \) due to constrain the weight, we calculate the number of edges in the graph whose weight is greater than \( t \).

And \( e_{\text{group}} \) is the error caused by grouping:
\[
e_{\text{group}}(G, \Omega_{(\theta,t)}) = \sum_{s \in \Omega_{(\theta,t)}} \sum_{h \in s} \left| h - \frac{\sum_{k \in s} h_k}{|s|} \right|,
\]  
(3)

where \( h \) represents the number of nodes with strength \( s \) of node strength histogram after preprocessing.

Last error \( e_{\text{noise}} \) is caused by adding Laplace noise.

Finally, in order to apply the exponential mechanism, we give the Lemma 1 to prove the global sensitivity of \( q(G, \Omega_{(\theta,t)}, e_i) \) is \( 10\theta_{\text{max}} + 4 \).

**Lemma 1.** For any two neighboring datasets \( G \) and \( G' \), we have:
\[
\left| q(G, \Omega_{(\theta,t)}, e_i) - q(G', \Omega_{(\theta,t)}, e_i) \right| \leq 10\theta_{\text{max}} + 4
\]

**Proof.** Supposing graph \( G \) has one more node \( \hat{v} \) than graph \( G' \). Base on [5], we have
\[
\left| e_p(G, \theta) - e_p(G', \theta) \right| \leq 2(\deg(\hat{v}) + 1) \leq 2(\theta_{\text{max}} + 1)
\]
and
\[
\left| e_{\text{group}}(G, \Omega_{(\theta,t)}) - e_{\text{group}}(G', \Omega_{(\theta,t)}) \right| \leq 4\theta + 2.
\]
And Laplace noise does not change with the datasets, so
\[
\left| e_{\text{noise}}(G, \Omega_{(\theta,t)}, e_i) - e_{\text{noise}}(G', \Omega_{(\theta,t)}, e_i) \right| = 0.
\]
In addition, adding a node \( \hat{v} \) to the graph will cause \( G \) to add \( \deg(\hat{v}) \) edges at most. Because the maximum degree of nodes in the compressed graph is \( \theta \), so:
\[
\left| e_o(G, \theta, t) - e_o(G', \theta, t) \right| = 4t \left[ \left| e \in E_p, \text{value}(e) > t \right| \right] \leq 4\theta
\]
To sum up, we have:
\[
\left| q(G, \Omega_{(\theta,t)}, e_i) - q(G', \Omega_{(\theta,t)}, e_i) \right| \leq 2(\theta_{\text{max}} + 1) + 4\theta + 4\theta + 2 \leq 10\theta_{\text{max}} + 4.
\]

### 4.4 Privacy analysis
In this section, we will theoretically analyze the privacy of the \((\theta, t)\)-Grouping-Hist algorithm.

**Lemma 2.** The \((\theta, t)\)-Grouping-Hist algorithm satisfies the node DP.

**Proof.** In the third step of the publication algorithm, the exponential mechanism is used to select the parameter combination that minimizes the error, and in this step, the privacy budget used is \(\varepsilon_2\); in the fifth step of the publication algorithm, the privacy budget \(\varepsilon_1\) is used in the Laplace mechanism to add Laplace noise to the grouped histogram, and finally output the noisy node strength histogram. It can be seen from the Sequential Composition of DP that the \((\theta, t)\)-Grouping-Hist algorithm satisfies \((\varepsilon_1 + \varepsilon_2)^{-1}\) node-DP.

5. **Experiment**

The following will compare the performance of \((\theta, t)\)-Grouping-Hist algorithm, T-Hist (Traditional Histogram) algorithm, and SER-Hist (SER-based Node Strength Histogram) algorithm through simulation experiments on four different datasets. The T-Hist algorithm was first proposed in [16], which does not do any preprocessing on graph data, and directly adds Laplace noise to each bucket of the histogram; SER-Hist algorithm is an algorithm modified in this paper. In [5], Zhang Applied SER to publish node degree histogram under node-DP.

5.1. **Experiment data**

The datasets of the simulation experiment come from the four real networks. These four datasets are downloaded from. The following table gives the information of these four data sets:

| datasets    | | | strength | weight | Degree |
|-------------|---|---|-----------|--------|--------|
| US_airport  | 2939 | 15677 | [1,1358]  | [1,11] | [1,242] |
| soc-sig-bit | 5881 | 21492 | [0,1890]  | [-20,20] | [1,795] |
| EIES        | 34  | 474  | [30,175]  | [1,8]  | [13,33] |
| social-Net  | 1899 | 13838 | [1,1546]  | [1,184] | [1,255] |

**Tab. 2** Information on datasets

5.2. **Parameter settings**

In this paper, the value range of privacy budget \(\varepsilon\) is [0.1, 2.0], and We divide \(\varepsilon\) into \(\varepsilon_1\) and \(\varepsilon_2\), where \(\varepsilon_1=0.9\varepsilon\), \(\varepsilon_2=0.1\varepsilon\). In addition, the degree threshold candidate set \(\Theta=[1,200]\) and the weight threshold candidate set \(T=[1,20]\). Due to the randomness of the noise, this paper chooses to conduct 30 experiments for each one, and finally takes the average of the 30 experiment results as the output.

5.3. **Results and analysis**

Referring to the metric in the paper [4], this article will use L1 error and KS distance as the measures of algorithm effectiveness. For any two distributions \(p\) and \(p'\), the L1 error between them is calculated by \(\|p-p'\|\), and if the lengths of \(p\) and \(p'\) are not the same, it is solved by adding 0; the KS distance is equal to \(\max_i |CDF_p(i) - CDF_{p'}(i)|\), where \(CDF_p(i)\) is the cumulative distribution function value of the node strength on the distribution \(p\).

Figure 1 and 2 show the L1 error and KS distance respectively, obtaining by three different algorithms.
The experimental results show that under the same privacy budget, the L1 error and KS distance of the T-Hist algorithm are the largest. This is because T-Hist has a high global sensitivity, which will cause too large noise. The SER-Hist preprocesses original graph through SER algorithm, which effectively reduces the sensitivity of the query function, so the L1 error and KS distance are smaller than the T-Hist algorithm. Finally, the \((\theta,1)\) -Grouping-Hist algorithm proposed in this paper performs significantly better than other two algorithms on different datasets. Because \((\theta,1)\) -Grouping-Hist not only reduces the global sensitivity within a feasible range, but also reduces the accumulative noise through the weight constraint and histogram grouping algorithm. Experimental results also show that for datasets with sparse degree distribution and weight distribution, the error of \((\theta,1)\) -Grouping-Hist is smaller, because there are fewer nodes with a degree greater than \(\theta\) and fewer edges with a weight greater than \(t\) in a graph, then more information of graph can be retained.

In general, \((\theta,1)\) -Grouping-Hist is superior to other two algorithms. It also shows that \((\theta,1)\) -Grouping-Hist proposed in this paper provides a higher level of privacy protection while ensure the utility of graph data.

6. Conclusion
In this paper, we have studied how to accurately publish the node strength histogram of weighted graph under node-DP. We reduce the global sensitivity of query function and the accumulative noise of long-range queries by preprocessing, and use the HCG algorithm to group the buckets with close values in histogram into one group. Meanwhile, in order to improve the accuracy of publishing the node strength histograms, we have proposed the \((\theta,1)\) -Grouping-Hist algorithm. The experimental results show that the \((\theta,1)\) -Grouping-Hist algorithm not only satisfies node-DP but also guarantees the utility of graph data. In the future, we plan to improve the efficiency of \((\theta,1)\) -Grouping-Hist algorithm by reducing the time complexity of grouping algorithm.

Acknowledgments
This work was supported in part by the National Natural Science Foundation of China under no. 61862011, Guangxi Science and Technology Foundation nos. 2018GXNSFAB138116 and 2019GXNSFGA245004.

References
[1] Dwork C 2006.Proc. of the 33rd Int. Conf. on Automata, Languages and Programming Springer pp 1-15
[2] Ping X, TQ Z and XF W 2014 A survey on differential privacy and applications J.Jisuanji Xuebao/Chinese Journal of Computers pp 101-122
[3] Michael H, Chao L, Gerome M and David J 2009 Accurate estimation of the degree distribution of private networks 2009 Ninth IEEE Int Conf. on Data Mining pp 169-178
[4] Weiyen D, Ninghui L and Min L 2016 Publishing graph degree distribution with node differential privacy Proc. of the 2016 Int. Conf. on Management pp 123-138
[5] Yuxuan Z, Jianghong W and Ji L 2019 Graph degree histogram publication method with node-differential privacy J.Journal of Computer Research and Development p 508
[6] Qing Q, Zhixu L and Pengpeng Z 2018 Publishing graph node strength histogram with edge differential privacy Int. Conf. on Database Systems for Advanced Applications (Springer) pp 75-91
[7] Kasiviswanathan S, Nissim K, Raskhodnikova S and Smith A 2013 Analyzing graphs with node differential privacy Theory of Cryptography Conf. (Springer) pp 457-476
[8] Blocki J, Blum A, Datta A and Sheffet O 2013 Differentially private data analysis of social networks via restricted sensitivity Pro. of the 4th Conf. on Innovations in Theoretical Computer Science pp 87-96
[9] Chatzikokolakis K, Andrés M.E and Bordenabe N.E 2013 Broadening the scope of differential privacy using metrics Int. Symposium on Privacy Enhancing Technologies Symposium (Springer) pp 82-102
[10] Andrés,M.E, Bordenabe, N.E, Chatzikokolakis K and Palamidessi C 2013 Geo-indistinguishability Differential privacy for location-based systems Pro. of the 2013 ACM SIGSAC Conf. on Computer & communications security pp 901-914
[11] Nissim K, Raskhodnikova S and Smith A 2007 Smooth sensitivity and sampling in private data analysis Pro.J.of the thirty-ninth annual ACM symposium on Theory of computing pp75-84
[12] Xiao Y, Xiong L and Yuan C 2010 Differentially private data release through multidimensional partitioning Jrkshop on Secure Data Management(Springer) pp 150-168
[13] Xu J, Zhang Z and Xiao X 2013 Differentially private histogram publication J.The VLDB Journal pp 797-822
[14] Qardaji W, Weining Y and Ninghui L 2013 Understanding hierarchicall methods for differentially private histograms J.Pro. of the VLDB Endowment pp 1954-1965
[15] Xiaojian Z, Rui C and Jianliang X 2014 Towards accurate histogram publication under differential privacy Pro. of the 2014 SIAM Int. Conf. on data mining(SIAM) pp 587-595
[16] Dwork C, Roth A 2014 The algorithmic foundations of differential privacy J.Foundations and Trends in Theoretical Computer Science pp 211-407