Research on Social Networks Publishing Method Under Differential Privacy

Han Wang(✉) and Shuyu Li

School of Computer Science, Shaanxi Normal University, Xi'an 710119, China
{wanghan, lishuyu}@snnu.edu.cn

Abstract. Data publishing for large-scale social network has the risk of privacy leakage. Trying to solve this problem, a differential private social network data publishing algorithm named DP-HRG is proposed in the paper, which is based on Hierarchical Random Graph (HRG). Firstly, the social network is divided into 1-neighborhood subgraphs, and the HRG of each subgraph is extracted by using both Markov Monte Carlo (MCMC) and exponential mechanism to compose the HRG candidate set. Then an average edge matrix is obtained based on the HRG candidate set and perturbed by a random matrix. Finally, according to the perturbed average edge matrix, a 1-neighborhood graph is regenerated and pasted into the original social network for publishing. Experimental results show that the proposed algorithm preserves good network characteristics and better data utility while satisfying the requirement of privacy protection.

Keywords: Differential privacy · Social network · Hierarchical random graph · Data publishing · Privacy protection

1 Introduction

With the rapid development of mobile Internet, social networks have become an increasingly significant way of communication among people. Social networks contain massive valuable information about individual and social relationships. Meanwhile, such information usually contains sensitive information. For example, in a medical network, the communication between an AIDS patient and a doctor may be considered sensitive information. The direct release and analysis of such information may violate individual privacy and cause extremely serious consequences. Therefore, it has become a very important issue about how to ensure the effective release of information in social network without leaking individual privacy. Traditional privacy protection methods, such as k-anonymity [1], l-diversity [2] and t-closeness [3], have been extensively used in practical applications. The basic idea of these methods is to employ the techniques of de-identification, generalization and suppression to process the attributes and records in the original data set to satisfy anonymity requirements and finally release the anonymous data set. However, all these methods are related to the background knowledge of potential attackers and privacy quantization cannot be strictly verified. Dwork [4] proposed the differential privacy theory in 2006, which successfully solved the problems encountered by traditional privacy protection methods. Differential privacy method does not need to consider the background knowledge of attackers, and provides

J. Li et al. (Eds.): SPNCE 2019, LNICST 284, pp. 58–72, 2019.
https://doi.org/10.1007/978-3-030-21373-2_6

a strict definition on privacy protection and provides a quantitative evaluation method. Differential privacy has become a research hotspot in the field of privacy protection.

Differential privacy is a data-distortion-based privacy protection technology that uses a noise-adding mechanism to distort sensitive data while ensuring data availability. However, due to the complexity of social networks, the direct addition of noise will probably result in a significant decline in the utility and values of social network data. Therefore, this research aims to preserve the original characteristics of social network and release the disturbed information as much as possible under the condition of satisfying the requirements for privacy protection.

2 Preliminaries

2.1 Differential Privacy

Definition 1 (ε-differential privacy). In proximate data sets D_1, D_2 (one and only one piece of data is different in the two data sets), the algorithm M can output any $S \subseteq Range(K)$ that satisfies

$$\Pr[K(D_1) \in S] \leq e^{\varepsilon} \times \Pr[K(D_2) \in S] \tag{1}$$

The algorithm is said to satisfy differential privacy protection.

Definition 2 (Gaussian mechanism). For a query function $f : D \to R^d$ on the given data set D, let $\sigma = \Delta_2 f \sqrt{2\ln(2/\delta)}/\varepsilon$, $N(0, \sigma^2)$ is an independent identically distributed Gaussian random variable (i.e. Gaussian noise), then the random algorithm M : $M(D) = f(D) + (N_1(0, \sigma^2), N_2(0, \sigma^2) \ldots N_d(0, \sigma^2))$ provides (ε, δ) differential privacy. The Gaussian mechanism is suitable for processing numerical data.

Definition 3 (Exponential mechanism). Given a scoring function $u(D, r)$, for an input data set D, the output is a random algorithm M of entity object $r \in Range$. Let Δq be the sensitivity of the function $u(D, r)$. If the algorithm M chooses and output r from $Range$ with a probability proportional to $\exp\left(\frac{\varepsilon \cdot u(D,r)}{2\Delta q}\right)$, then the random algorithm M provides ε-differential privacy. The exponential mechanism is suitable for processing non-numeric data.

2.2 Social Network

Undirected and unweighted graph $G(V, E)$ is used to model a social network, where V is a point set, E is an edge set and $|V|$ represents the number of nodes.

Definition 4 (1-neighborhood graph). For each node in V, if the social network, if there is a sub-graph consisting of only 1-hop node and v itself in the social network, then we call this sub-graph as the 1-neighborhood graph centered on node v, and it can be labeled as $g(v)$.

For example, Fig. 1 is a social network containing 9 nodes. Figure 2 shows the two 1-neighborhood graphs centered on nodes D and F, respectively, in a social network.

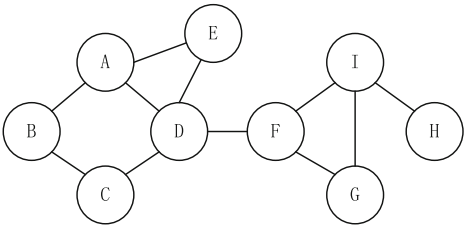
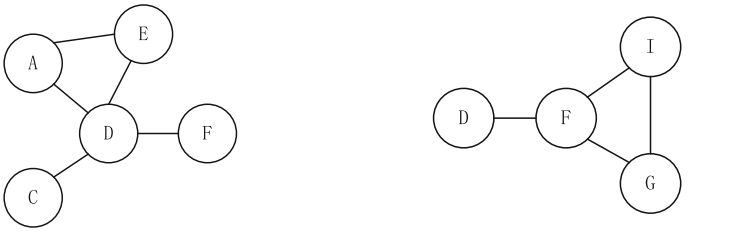


Fig. 1. Social network



(a) 1-neighborhood graph of node D (b) 1-neighborhood graph of node F

Fig. 2. 1-neighborhood graph of social network G

2.3 Hierarchical Random Graph

Clauset et al. [5] proposed the concept of hierarchical random graph (HRG) in 2008. HRG represents a social network using network hierarchy and a set of link probabilities, to seek the best hierarchical graph of the network, thus formulating a more accurate description of the hierarchical structure. The hierarchical structure of HRG is represented by leaf nodes and a tree diagram T consisting of internal nodes. The leaf nodes represent the real nodes in social network. The internal nodes represent the probability P_r that there is an edge between left subtree L_r and right subtree R_r, where r is the root and P_r is the degree of relation between different groups. The larger the degree of relation, the closer the relationship is between the two. P_r is defined by

$$P_r = \frac{e_r}{(n_{L_r} \cdot n_{R_r})} \tag{2}$$

where e_r represents the number of edges that exist between the left and right subtrees of internal node r, n_{L_r} represents the number of nodes in the left subtree, and n_{R_r} represents the number of nodes in the right subtree.

Similarity measure L is used to indicate whether the HRG of social network retains the structural attributes of the original social network to the greatest extent. Similarity measure is defined as

$$L(T, \{p_r\}) = \prod_{r \in T} p_r^{e_r} (1 - p_r)^{n_{L_r} n_{R_r} - e_r} \tag{3}$$

For a given T, the similarity measure represents the product of the link probabilities of all internal nodes. In this paper, the calculation process is simplified by taking the logarithm of the similarity measure L, shown as below:

$$\log L(T, \{p_r\}) = -\sum_{r \in T} n_{L_r} n_{R_r} h(p_r) \tag{4}$$

where $h(p_r) = -p_r \log p_r - (1 - p_r) \log(1 - p_r)$ is the Gibbs-Shannon entropy function. The higher the similarity, the better the structure of social network can be described. For convenience, $\log L(T)$ is used to replace $\log L(T, \{p_r\})$ in the rest sections.

3 Related Work

Many achievements have been made in applying differential privacy to social networks. Wang et al. [6] proposed a RescueDP method for the problem of the publishing of real-time spatio-temporal data in social networks. This method provided privacy-preserving statistical data publishing on infinite time stamps by integrating adaptive sampling and privacy budget allocation, dynamic grouping, disturbance and filtering technologies. The RescueDP method improved the practicability of real-time data and had strong privacy protection. Li et al. [7] proposed a network weight-based privacy protection algorithm. The algorithm treated the edge weight sequence of the undirected weighted graph as a non-attribute histogram, and added the weight containing sensitive information to the Laplace noise to meet the requirements for differential privacy. To reduce the amount of noise, the buckets with the same count in the histogram were merged into groups, and the requirements for differential privacy were satisfied through the inter-group k-unidentifiability. The reasoning of consistency was performed on the original weight sequence to keep the shortest path of the network unchanged. Tian et al. [8] proposed the DWDPP method for privacy protection in weighted social network. The method used discrete wavelet transform to decompose the weight matrix, and then added Laplace noise to the high-frequency detail matrix of each layer and the low-frequency approximation matrix of the last layer to reconstruct the weight matrix. Data availability was preserved by this method. Xiao et al. [9] modeled the social network graph by HRG, then added noise to the generated graph, and finally restored the HRG with noise to obtain a network graph that satisfied the differential privacy. However, the time complexity of the algorithm was high. Most of the above algorithms are directly processing a whole social network graph. So, for a large-size social network graph, these algorithms may encounter the problems such as long processing time and sharp

decline in data availability. Therefore, based on the privHRG algorithm proposed by Xiao [9], this paper proposes a social network publishing algorithm DP-HRG which satisfies the differential privacy requirements for undirected and unweighted social networks, as well as preserves the features of original network. This research mainly contains the following work.

- For the problems of large size and data correlation in real social network graphs, this paper proposes a 1-neighborhood graph partitioning method based on the largest independent set of social network. This method can effectively reduce the network size and improve the computational efficiency.
- When searching the best matching tree in the candidate tree set, the sampling technique that combines Markov Monte Carlo (MCMC) method and exponential mechanism is designed to improve the sampling efficiency while protecting the privacy. In addition, a subgraph re-generation and link method is proposed to paste the regenerated 1-neighborhood graph into the original social network to publish the complete social network.
- The DP-HRG algorithm is experimentally evaluated on two kinds of real social network datasets, and the privacy analysis, sensitivity analysis and utility analysis of the algorithm are carried out.

4 Social Network Publishing Algorithm Based on Differential Privacy

4.1 The Ideas of the Algorithm

For small-size social networks, we can construct their HRG directly and then perform the corresponding disturbance operations. However, social network usually has large size, their nodes and structures are complex, and various complex factors affect each other, which is very inefficient to directly construct HRG. In this case, we mainly face two problems: (1) how to construct HRG efficiently for large-size social networks and perform disturbance; (2) how to find the best matching tree from the candidate tree set while satisfying privacy protection.

For the first problem, a partition-based method is used to improve efficiency. On the basis of the largest independent set, the original social network is partitioned into several 1-neighborhood graphs, and HRGs are constructed for these subgraphs, and then corresponding disturbances are performed. On the one hand, the graph size is greatly reduced, and the spatial size of the output HRG is reduced. Besides, since the generation of the largest independent set is uncertain, the attacker will not know which subgraphs are disturbed. On the other hand, the 1-neighborhood graph itself reflects the local characteristics of the network. The method of constructing HRG on subgraphs and conducting the disturbance makes the noise is added to only a part of the subgraphs in the whole social network, which preserves the availability of the social network to a greater extent. In addition, 1-neighborhood graph can represent the direct relation between a target user and all of its neighbors, and it needs to be protected. For the second

problem, MCMC is employed in the sampling process to improve the sampling efficiency, and the exponential mechanism is incorporated to satisfy the differential privacy.

In order to guarantee the greater accuracy and better availability of the sampled sample tree set, each HRG in the sample tree set is converted into a corresponding edge number matrix, and obtain the average edge number matrix of all edge number matrixes. Then, for the disadvantage of insufficient privacy protection, the average edge number matrix is disturbed by random matrix method, and only a small amount of noise is required to ensure differential privacy, thereby improving data availability.

The DP-HRG algorithm can be divided into the following four steps:

1. Find the largest independent set of social networks and obtain the 1-neighborhood graph for each node in the largest independent set;
2. Extract the HRG of each 1-neighborhood graph using MCMC method and exponential mechanism to obtain a sample tree set;
3. Convert the HRG in the sample tree set to the edge number matrix, and then obtain the average edge number matrix which is then disturbed using use the random matrix;
4. Regenerate the 1-neighborhood graphs and paste them into the original social network, and finally publish the disturbance graph;

The algorithm framework is as follows:

Table 1. DP-HRG algorithm

Algorithm 1 DP-HRG algorithm
Input: original social network G, privacy budget ε, random noise variable σ
Output: disturbed social network $\tilde{G}$

1. Obtain the 1-neighborhood graph of the largest independent set;
2. Perform sampling to obtain sample tree set S $\leftarrow$ SampleHRGs $(g(v),\varepsilon)$;//Algorithm 2
3.1. Convert the HRGs in S into edge number matrix and put them into a set M ;
3.2. Obtain the average edge number matrix after disturbance $\tilde{M} \leftarrow$RandMatrixDisturb (M,σ);//Algorithm 3
4. Subgraph generation and link $\tilde{G} \leftarrow$SubgraphGenAndLink $(G,\tilde{M})$;//Algorithm 4
5. Publish the disturbed social network $\tilde{G}$;

4.2 Algorithm Design

Obtaining the Largest Independent Set and 1-Neighborhood Graph

The algorithm needs to partition the original social network into several subgraphs and disturb some of the subgraphs. However, due to the correlation between nodes in the social network, the social network cannot be directly partitioned. Therefore, it is required to find the largest independent set of the social network *G*, and then obtain the

1-neighborhood graph of each node in the largest independent set. This method not only reduces the size of the original network, but also preserves the correlation, and also lays the foundation for data disturbance.

Extracting the HRG to Obtain Sample Tree Set

Due to the nature of the HRG, each 1-neighborhood graph corresponds to multiple HRGs. Let T be a set of all possible trees, so for a network with $|V|$ nodes, the number of T is $|T| = (2|V| - 3)!!$, where !! represents a semi-factorial symbol. When the size of a social network is large, the efficiency of finding the best HRG from many HRGs will be greatly reduced, although the partition is employed to reduce the output space, the output space is still large for a larger subgraph. Therefore, the MCMC method is used to control time complexity and reduce computation time. At the same time, the exponential mechanism is incorporated with MCMC method. The similarity measure function is used as the scoring function u. The acceptance probability of the MCMC process is changed from $\frac{u(T)}{u(T')}$ to $\alpha = \frac{\exp\left(\frac{\varepsilon}{2\Delta q}u(T')\right)}{\exp\left(\frac{\varepsilon}{2\Delta q}u(T_{i-1})\right)}$, where Δq is the global sensitivity of the scoring function u.

This algorithm first selects an arbitrary tree as the initial state of the Markov chain, and then performs looping executions: randomly selects the neighbor tree T' of T_{i-1} and updates it as follows:

$$T_i = \begin{cases} T' & \text{with probability } \alpha \\ T_{i-1} & \text{with probability } 1 - \alpha \end{cases} \tag{5}$$

Therefore, if we need randomly select the neighbor tree T' of T_{i-1}, we first need to randomly select the internal nodes r other than the root node from T_{i-1}, and find their brothers and two children, and then transform the three subtrees to generate the two alternative trees of r, as shown in Fig. 3. One of the two alternative trees is selected as the neighbor tree T'.

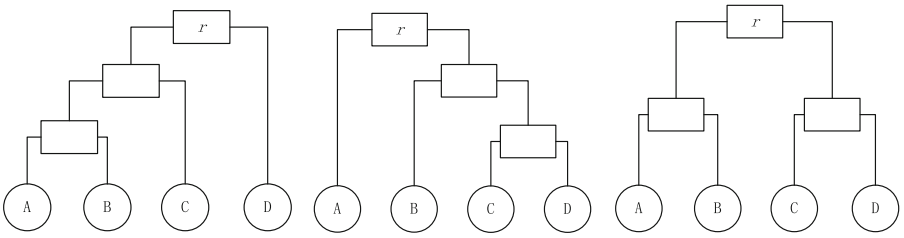


Fig. 3. Three structures of subtrees

When the MCMC process reaches a steady state, the sampling is performed at regular intervals to obtain a set S containing N sample trees.

Table 2. HRGs sampling of differential privacy

Algorithm 2: HRGs sampling of differential privacy (SampleHRGs)
Input: 1-neighborhood graph $g(v)$, privacy budget ε
Output: the set $S=\{T_1,T_2,...,T_n\}$ containing N sample trees
1. Randomly select a tree T_0 to initialize the Markov chain; 2. For the every step i of Markov chain, perform looping execution: Randomly select an internal node r; Randomly select a subtree of r as the neighbor tree T_i' of T_{i-1}; Take the T_i' as T_i with the acceptance probability of $\min(1,\frac{\exp(\frac{\varepsilon}{2\Delta q}u(T'))}{\exp(\frac{\varepsilon}{2\Delta q}u(T_{i-1}))})$; When the Markov chain reaches steady state, the loop ends; 3. Select a number of sampling trees in the generated sample tree set; 4. Return to the sample tree set S;

Random Matrix Disturbance

After obtaining the sample tree set S, we need to convert the N sample trees in S into edge number matrix and perform further disturbance to achieve stronger privacy protection. The combination of random matrix theory and differential privacy is adopted. Firstly, the N sample trees in the sample tree set are transformed into N edge number matrixes, and their average $\bar{M}$ is obtained. Then, the average edge number matrix is disturbed by the Gaussian random noise matrix, and finally we obtain the average edge number matrix with random disturbance.

The horizontal and vertical coordinates of the edge number matrix represent the coordinates of the nodes in the social network, and the values of these elements represent the number of linked edges of the left and right subtrees of the internal node r. After disturbing edge number matrix using the random matrix, a new average edge number matrix with random disturbance is obtained, and the number of edges after the disturbance is labeled as $\tilde{e}_r$.

Table 3. Random disturbance matrix

Algorithm 3: Random disturbance matrix (RandMatrixDisturb)
Input: Edge number matrix set M, random noise variable σ
Output: Average edge number matrix after disturbance $\tilde{M}$
1. Obtain the average edge number matrix obtained by averaging the N edge number matrixes in M; 2. Obtain the random disturbance matrix by sampling Gaussian distribution $N\sim(0,\sigma^2)$; 3. Calculate the average edge number matrix with random disturbances $\tilde{M}=\bar{M}+Q$;

Subgraph Generation and Link

After obtaining the average edge number matrix with random disturbance, the probability $\tilde{P}_r$ of each pair of leaf nodes corresponding to the disturbance is calculated based on the value of $\tilde{e}_r$, and the edge is placed between the pair of leaf nodes to generate the disturbed subgraph $\tilde{g}(v)$. The v nodes in the undisturbed graph are randomly determined in the disturbed graph $\tilde{g}(v)$, and then comparison is performed between the disturbed subgraph and the undisturbed subgraph. If an edge is added or deleted between the nodes of the disturbed subgraph, modification should be made in the corresponding 1-neighborhood graph in the original social network. In this way, the disturbed subgraph can be pasted into the original social network.

For example, Fig. 4 is the 1-neighborhood graph after the disturbance on Fig. 2(b), and then it is compared with the 1-neighborhood graph of node F in Fig. 1. In the disturbed graph, the D node is linked to the nodes F and G with edges, while in the original graph, the node D is only linked to the node F, so it is required to add an edge between D and G in the original graph. The other three nodes F, G, and I all take the similar operations, and finally we obtain a disturbed social network graph, as shown in Fig. 5.

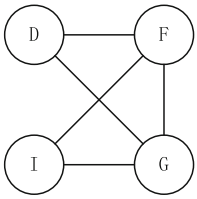


Fig. 4. 1-neighborhood graph after disturbance

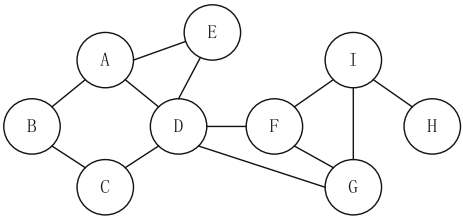


Fig. 5. Social network after disturbance

4.3 Privacy Analysis

The Algorithm 2, which combines MCMC with exponential mechanism to extract HRG, is essentially a method of sampling the output with a probability proportional to $\exp(\varepsilon u(D, r)/2\Delta q)$ in the target distribution, where $u(D, r)$ is the scoring function and Δq is the sensitivity. Therefore, by matching the smooth distribution of the MCMC with the target distribution required for the exponential mechanism, the MCMC can be

Table 4. Subgraph generation and link

Algorithm 4: Subgraph generation and link (SubgraphGenAndLink)
Input: Social network G, average edge number matrix with disturbance $\tilde{M}$
Output: distributed social network $\tilde{G}$
1. For every $g(v)$ in G, perform looping execution: 1.1. For every pair of nodes i, j in $g(v)$, perform looping execution: (1) Find the number of linked edges $\tilde{e}_r$ of the left and right subtrees of the internal node r corresponding to each pair of leaf nodes in the average edge number matrix with disturbance; (2) Calculate the link probability $\tilde{P}_r$ after the disturbance based on $\tilde{e}_r$; (3) Place the edges between the i, j in network graph with probability $\tilde{P}_r$ to gen er ate a disturbed subgraph; End of loop 1.2 Randomly determine the v nodes in the undisturbed graph $g(v)$ in the disturbed graph $\tilde{g}(v)$; 1.3 For each pair of nodes i, j in $\tilde{g}(v)$, perform looping execution: (1) If there is an edge between i and j, then add an edge between them in $g(v)$; (2) If there is no edge between i and j, then delete the edge between them in the $g(v)$ of G; End of loop End of loop 2. Return to the disturbed social network graph;

used to implement the exponential mechanism. The scoring function $u(T)$ of T is set as $\log L(T)$, then the acceptance probability of MCMC is given by

$$\min\left(1, \frac{\exp\left(\frac{\varepsilon}{2\Delta u} \cdot \log L(T')\right)}{\exp\left(\frac{\varepsilon}{2\Delta u} \cdot \log L(T_{i-1})\right)}\right) \tag{6}$$

When the Markov chain converges to steady state, the sample is extracted from the probability mass function which is expressed as

$$\Pr(T) = \frac{\exp\left(\frac{\varepsilon}{2\Delta u} \cdot \log L(T)\right)}{\sum_{T' \in T} \exp\left(\frac{\varepsilon}{2\Delta u} \cdot \log L(T')\right)} \tag{7}$$

This means that the exponential mechanism outputs T with a probability proportional to $\exp\left(\frac{\varepsilon}{2\Delta u} \cdot \log L(T)\right)$. Therefore, Algorithm 2 satisfies differential privacy.

The random matrix disturbance process of Algorithm 3 satisfies the differential privacy, which has been proved in literature [10]. The rest steps of the algorithm do not consume the privacy budget. According to the sequenced combination property of differential privacy, the algorithm as a whole satisfies 2ε-differential privacy.

4.4 Sensitivity Analysis

Sensitivity can be expressed as:

$$\Delta q = \max(u(T(e_r)) - u(T(e_r - 1)))$$
$$\log(\Delta q) = \max\left(n_{L_r} n_{R_r}\left(h\left(\frac{e_r}{n_{L_r} n_{R_r}}\right) - h\left(\frac{e_r - 1}{n_{L_r} n_{R_r}}\right)\right)\right)$$

Let $N = n_{L_r} n_{R_r}$, then

$$\Delta q = \log N - (N-1) \cdot \log \frac{N-1}{N} =$$
$$\log N + (N-1) \log\left(1 + \frac{1}{N-1}\right) =$$
$$\log N + \log\left(1 + \frac{1}{N-1}\right)^{N-1} <$$
$$\log N + \log e \leq \log \frac{|V|^2}{4} + 1 = O(\log n)$$

When $n_{L_r} \cdot n_{R_r}$ is increasing, Δq is monotonically increasing. When n_{L_r} equals n_{R_r} and the value is equal to half of the number of all nodes, i.e. $|V|/2$ ($|V|$ represents the number of nodes), Δq reaches its maximum. When the number of nodes increases, the magnitude of the noise increases, where both $|V|$ and n represent the total number of nodes in the graph. Therefore, the sensitivity of the method is $O(\log n)$. The specific proof process can be found in literature [9].

5 Experiment and Results Analysis

5.1 Experimental Setup

The experiment adopted two real data sets from SNAP [11]: the polblogs data set and the facebook data set. The statistics of the data sets used in the experiment are shown in Table 5. The experimental environment was Intel Xeon, CPU of E7-4830 2.13 GHz, RAM of 32 GB, operating system of Windows Server 2008, and the algorithm is written in C++ language.

Table 5. Data set information statistics table

Data set	Number of nodes	Number of edges
polblogs	1490	19090
facebook	4039	88234

5.2 Utility Analysis

The algorithm proposed in this paper was compared with the algorithm of literature [8] to verify the utility of the proposed algorithm. In the experimental renderings, Origin represents the original social network, privHRG represents the algorithm proposed in literature [8], and DP-HRG represents the algorithm proposed in this paper. The utility of the proposed DP-HRG algorithm was examined by comparing it with Origin and the network graph processed by privHRG from the aspects of node degree distribution, average clustering coefficient and shortest path length distribution. Due to the randomness of the algorithm, three tests were carried out for each aspect to obtain an average. When conducting the experiments, we set a relatively large privacy budget and variance, with the privacy budget $\varepsilon = 1$ and variance $\sigma = 1$. The reason for this is as follows. On the one hand, the generation of HRG in this algorithm and the randomness of the subgraph generation and link process had rendered the algorithm certain privacy protection capabilities; on the other hand, the structural characteristics of the complex graph itself determined that a relatively large privacy budget could guarantee the privacy protection effect.

Degree Distribution: The degree of a node in a network refers to the number of links that the node has with other nodes. The degree of node distribution reflects the structure of a network to a certain extent. Figures 6 and 7 show the results of node degree distribution under different data sets. In order to better represent the experimental results, we intercepted the parts of the two graphs with significant changes in degrees for better illustration. It can be seen from the experimental results that both DP-HRG and privHRG algorithms followed the overall trend that the larger the degree of nodes, the fewer the number of nodes in the original network, and they both maintained the original network structure characteristics for large social networks. Compared with the publishing results of the polblogs data set, those of the facebook dataset were closer to the original network, indicating that the good effect of DP-HRG algorithm for large social networks.

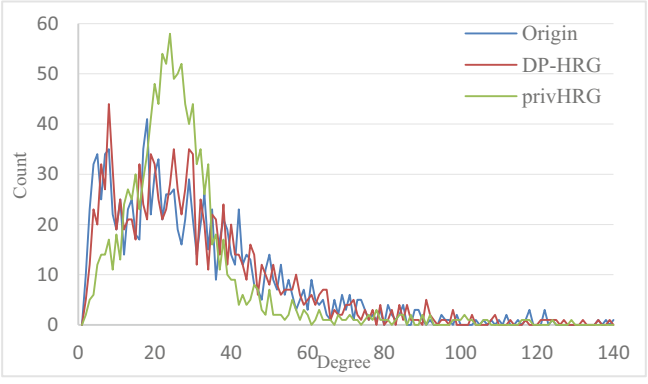


Fig. 6. Degree distribution of polblogs

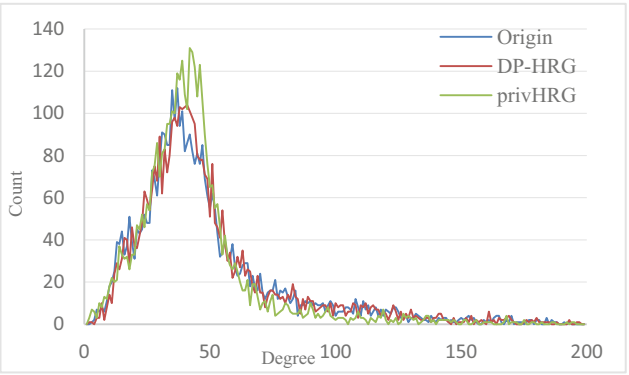


Fig. 7. Degree distribution of facebook

Average clustering coefficient: The clustering coefficient refers to the degree of aggregation of nodes in a network. It is assumed that node i in a network has k_i edges to link it with other nodes, then the clustering coefficient is $C_i = 2E_i/(k_i(k_i - 1))$, where E_i refers to actual number of edges that exist among k_i nodes. Average clustering coefficient is the average of the clustering coefficients C_i of all nodes i. Figure 8 shows the average clustering coefficients for two data sets under Origin, DP-HRG, and privHRG. It can be observed from the figure that DP-HRG and privHRG could maintain the clustering characteristics of the original network compared with the average clustering coefficient of the original network, but the average clustering coefficient of the network published by the DP-HRG algorithm was closer than that of the privHRG algorithm, with an error of less than 0.01 for the polblogs dataset, and an error of 0.0005 for the facebook dataset. This indicated that the DP-HRG algorithm could better maintain the aggregation characteristics of the nodes in original social network, and better describe the structure of the network, with even more obvious effect on large social networks.

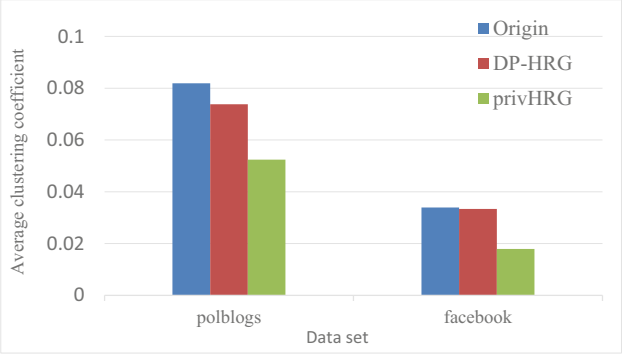


Fig. 8. Average clustering coefficient

Shortest path length distribution: This term refers to the distribution of the shortest path lengths between nodes in a network, which to some extent reflects the characteristics of a graph. Figures 9 and 10 show the shortest path length distribution of the two data sets. According to the experimental results, the number of paths with a length of 2 in the polblogs data set was the largest, and that with a length of 3 in the facebook data set was the largest. Both DP-HRG and privHRG algorithms were be able to preserve the characteristics of the original network path length. It should be noted that although the differences in some stages of the figure were not obvious, there would still be large differences between them for massive data.

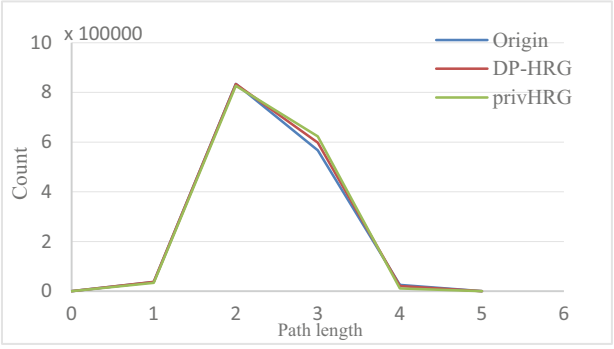


Fig. 9. Shortest path length distribution of polblogs

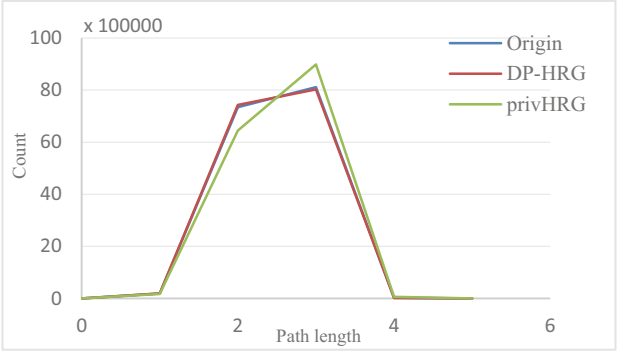


Fig. 10. Shortest path length distribution of facebook

6 Conclusions

This paper investigates the differential privacy-based social network publishing method using HRG. Firstly, on the basis of obtaining the largest independent set of a network and the 1-neighborhood graph, the method that combines MCMC method and exponential mechanism is employed to sample the HRG. Then, random matrix is used to

disturb the average edge matrix to further enhance the privacy protection. After that, the 1-neighborhood graph is generated. Finally, the new 1-neighborhood graph is pasted into the original social network for publishing. The experimental results show that the proposed algorithm guarantees good data utility under the premise of satisfying differential privacy. However, in the process of random matrix disturbance, due to the addition of noise to the number of edges, a great impact will still be formulated on the probability of the existence of HRG edge. In the future work, more efforts will be made to further optimize the mechanism of adding noise.

Acknowledgments. This work is supported by the Fundamental Research Funds for the Central Universities (No. GK201703055, No. GK201801004), CERNET Innovation Project (No. NGII20170703) and Key Research and Development Program of Shaanxi Province (No. 2017GY-064).

References

1. Sweeney, L.: k-anonymity: a model for protecting privacy. Int. J. Uncertainty Fuzziness Knowl. Based Syst. **10**(05), 557–570 (2002)
2. Machanavajjhala, A., Kifer, D., Gehrke, J.: L-diversity: privacy beyond k-anonymity. In: International Conference on Data Engineering, p. 24. IEEE, Atlanta (2006)
3. Ostrovsky, R., Yung, M.: How to withstand mobile virus attacks. In: Proceedings of the Tenth Annual ACM Symposium on Principles of Distributed Computing, pp. 51–59. ACM (1991)
4. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 1–12. Springer, Heidelberg (2006). https://doi.org/10.1007/11787006_1
5. Clauset, A., Moore, C., Newman, M.E.J.: Hierarchical structure and the prediction of missing links in networks. Nature **453**(7191), 98 (2008)
6. Wang, Q., Zhang, Y., Lu, X., et al.: Real-time and spatio-temporal crowd-sourced social network data publishing with differential privacy. IEEE Trans. Dependable Secure Comput. **15**(4), 591–606 (2018)
7. Li, X., Yang, J., Sun, Z., et al.: Differential privacy for edge weights in social networks. Secur. Commun. Netw. **2017**(4), 1–10 (2017)
8. Tian, H., Liu, J., Shen, H.: Diffusion wavelet-based privacy preserving in social networks. Comput. Electr. Eng. **67**, 509–519 (2018)
9. Xiao, Q., Chen, R., Tan, K.L.: Differentially private network data release via structural inference, pp. 911–920. ACM (2014)
10. Ahmed, F., Jin, R., Liu, A.X.: A random matrix approach to differential privacy and structure preserved social network graph publishing. In: Computer Science (2013)
11. SNAP Homepage. http://snap.stanford.edu. Accessed 17 Nov 2018