An Efficient Differential Privacy Method with Wavelet Transform for Edge Weights of Social Networks

Jun Yan[1,2], Hai Liu[3], and Zhenqiang Wu[1]
(Corresponding author: Zhenqiang Wu)

School of Computer Science, Shaanxi Normal University, Xi'an 710119, China[1]
School of Mathematics and Computer Applications, Shangluo College, Shangluo 72600, China[2]
Guizhou Provincial Key laboratory of Public Big Data, Guizhou University, Guiyang 550025, China[3]
Email: zqiangwu@snnu.edu.cn

(Received Mar. 12, 2021; Revised and Accepted Aug. 14, 2021; First Online Nov. 13, 2021)

Abstract

With more and more attention paid to privacy preservation in social networks, many effective methods based on differential privacy have been presented. To preserve the sensitive information of edge weights in a weighted social network, we combine the differential privacy with wavelet transform and devise a DPEW (Differential Privacy based Edge weight with Wavelet Transform) method. The proposed method satisfies differential privacy and provides better data utility. In this method, WTDP (Wavelet Transform based on Differential Privacy) algorithm can achieve differential privacy preserving while less noise is added on the edge weights. In addition, the properties of the original graph are maintained by EDU (Enhancing Data Utility) algorithm. The experimental results show that the DPEW method achieves ϵ-differential privacy and reduces the information loss of the edge weight than other methods.

Keywords: Differential Privacy; Perturbation Ratio; Wavelet Transform; Weighted Social Network

1 Introduction

Nowadays, with the widespread popularity of mobile Internet, the Internet has become more and more close to our life. For example, about ten years ago, we usually went out to shopping in the supermarket, but now we can buy almost anything online at home. With the help of Internet, we can use a mobile phone to record and analyze information about running and walking, which can promote us to keep training. Especially, the online social network, which merges our online and offline lives, has played a more and more important role in our daily lives. Through the Facebook platform, the largest social network in the world, which has 2 billion monthly active users, we can make friends and share information anytime and anywhere. In addition, we can also do many things on social networks, such as shopping, advertising, Video Live Broadcasting and so on, which bring great convenience to our lives. More importantly, recently, with the development of VSNs (Vehicular Social Networks), the numerous applications for VSNs will occupy our daily lives. Hence, we can say, social networks online have greatly changed our lifestyle.

However, social networks online appear to be a double-edged sword:although they bring us a lot of conveniences, they also present a great challenge to us. For example, social networks online contain a great quantity relationships between every individual, such as schoolmate relation, colleague relation and so on. Moreover, these relationships may be relevant to all kinds of attributes (weight values, directions), which are personal sensitive information. As a result, when social networks are published without privacy preserving, it is a great possibility to infer the hidden and secret information with high accuracy, which results in many privacy leakage problems. Therefore, in order to preserve privacy of social networks, it is critical for us to present effective privacy preserving methods.

For preserving privacy of social networks, we can abstract a social network as a graph where the vertices represent the individuals and the edges represent relationships among individuals. Therefore, the graph modified methods are widely used in this area. A simple method is naiVe anonymization method which is presented by Hay [6]. To resist connection-based attacks, edge and vertex modification methods which randomly perturbed the original graph are proposed. For example, Hay [6] also proposed the random perturbation algorithm. In addition, a random perturbation method called Blockwise random Add/Delete was developed by Ying [19]. In order to improve the data utility, many constrained perturbation methods to satisfy some desired constraints were developed, such as a spectrum preserving approach [20] and the k-anonymity model [12,14].

In addition to the methods mentioned above, a well-known privacy-aware computation method called differential privacy [3], which can defend against any attacks based on background knowledge, has been widely applied for privacy preserving in many areas, such as the smart grid information system [4]. Due to being able to provide rigorous privacy guarantee, the popular differential privacy mechanism has been used to publish sensitive graph data, such as the number of triangles and k-stars [16]. Different from adding noise to the graph data, the differential privacy technology based on the Stochastic Kronecker Graph Model [8] was introduced to provide privacy preserving on a graph, which can improve data utility significantly [13]. Thus, as a useful privacy preserving technology, differential privacy technology can also be widely applied in weighted social networks.

In weighted social networks, for preserving the weight value of edges which indicate the degree of intimacy between individuals, the researchers have proposed many methods which can be divide into two classes. One class is based on the K-anonymous technology [22], and the other is based on the differential privacy. Compared with the K-anonymous technology, the differential privacy will result in insufficient data utility because of a lot of Laplacian noise when preserving social networks. In order to improve the data utility, all kinds of transformation methods are used in differential privacy. As a special transformation method, the wavelet transform can not only provide rigorous privacy guarantee but also can keep a certain degree data utility, which is presented by Xiao [18].In this paper, a well known Haar wavelets, which has the simplest orthogonal basis among all discrete wavelet transforms, is used to achieve differential privacy while reducing the perturbation of noise. Further more, the shortest distances between some important nodes in the weighted social network are kept unchanged in our method, which make our method to have a better data utility than other methods.

In summary, our contributions are described as follows:

1) We devise a DPEW (Differential Privacy based Edge weight with Wavelet Transform) method, which satisfies the differential privacy with better data utility.

2) We propose two algorithms. The first algorithm is WTDP (Wavelet Transform based on Differential Privacy), which can achieve differential privacy preserving while adding less noise on the edge weights. The second is EDU (Enhancing Data Utility), which is an algorithm that can maintain the properties of original weighted social network to enhance the data utility.

3) We present the PR (perturbation ratio) to evaluate the different methods in privacy preserving, which is more intuitively than parameter ϵ. and we compare our method with other different methods in the synthesis and real data sets.

In the following sections, the organization of this paper is outlined as follows. In Section 2, we introduce many kinds of privacy preserving methods which are applied in social networks. we give some preliminaries, including the differential privacy, the wavelet transform and the properties of graph in Section 3. Section 4 describes our privacy preserving method and algorithms. The experimental results and comparison are illustrated in Section 5. Finally, Section 6 concludes this paper.

2 Related Work

Since differential privacy was put forward by C. Dwork, a lot methods based on differential privacy have been proposed, which can be classified into edge differential privacy and node differential privacy [7]. As one of the most important properties of a graph, the degree distribution was protected by an efficient algorithm based on K-edges differential privacy, which was provided by Hay [1]. In order to protect another important statistics, such as subgraph counts, Zhang [10] introduced a new method which guarantees differential privacy by using ladder framework. Comparing with edge differential privacy, node differential privacy could satisfy stronger privacy guarantees, but preform lower data utility. In order not to change original data significantly, Kasiviswanathan [5] use several techniques to develop node differential privacy algorithms, which improve the data utility. In the method based on node differential privacy [21], the aggregation technique and the cumulative histogram technique were used to obtain better data utility in publishing the degree distribution.

In weighted social networks, being a significant property, edge weighs can be protected by many techniques, such as K-anonymous technique and differential privacy technique. To prevent attacks based on background, k-anonymity of nodes method [15] and [k1, k2]-shortest path privacy method [17] have been presented. Based on differential privacy, a method with the MB-CI strategy is proposed to protect edge weight, which enhanced the accuracy and utility of the published data [9].

Due to having a better property on privacy preserving and data utility, the wavelet transform as a signal transformation method can be used for data perturbation. To prevent the privacy in certain data from being revealed in data mining, Liu [11] presented a method based on wavelet transform which maximized data utility. For better privacy, Xiao [18] achieved differential privacy by combining wavelet transform. In privacy preserving clustering, Dishabi [2] proposed a different privacy based method with daubechies-2 wavelet transform.

3 Preliminaries Knowledge

In this paper, a weighted social network is regarded as a simple, undirected, weighted graph G=(V, E, W), where

V=(v_1, v_2, $\cdots$, v_n) with each v_i representing an individual in social network, E=(e_1, e_2, $\cdots$, e_n) with each e_i describing a relationship between two v_i, W=(w_1, w_2, $\cdots$, w_n), each w_i describes a kind of attribute of e_i.

Definition 1. *(Neighboring graph). For two weighted graphs* G_1 =(V_1,E_1,W_1), G_2=(V_2,E_2,W_2), *if* $|V_1 \bigoplus V_2|+|E_1 \bigoplus E_2|=2$, *where* $\bigoplus$ *is Exclusive - OR operation, we can say that* G_1 *and* G_2 *are neighbors. Assuming* V_1=V_2, *if* $|E_1 \bigoplus E_2|=2$, G_1 *and* G_2 *are neighbors. In this paper, we assume that there are two different edges between two graphs* G_1 *and* G_2*. In general, because the difference of two graphs is two edges, edge differential privacy is used to achieve differential privacy.*

Definition 2. *(Differential Privacy). If a randomized algorithm* R *satisfies* ϵ*-differential privacy, there is a conclusion as following:*

$$P_r[\mathrm{R}(G_1) \in \mathrm{T}] \leq exp(\epsilon)P_r[\mathrm{R}(G_2) \in \mathrm{T}].$$

where T $\subseteq$ *Range(*R*),* G_1,G_2 *are neighbors and* ϵ *is a privacy budget. In order to achieve differential privacy, we comply with Laplace Mechanism to add the Laplace noise on the result of queries.*

Definition 3. *(Laplace Mechanism). In a weighted graph, assuming a query function is* Q*, where* G *is a weighted graph,* w *is a weight sequence of* G*. Given two* G_1 *and* G_2*, which are neighbors, according to the definition 1, the sensitive of* Q *is as following:*

$$\Delta Q = max_{G_1,G_2}\|Q(G_1) - Q(G_2)\|_1.$$

The Laplace mechanism is a special technique, which adds Laplace noise to the output of a query function to satisfy differential privacy.

$$R(G) = Q(G) + Lap(\frac{\Delta Q}{\epsilon}).$$

where the Laplace noise satisfies Laplace distribution, which is described as follows.

$$d(x) = \frac{1}{2b}exp(-\frac{|x-\mu|}{b})$$

where μ=0 b=$\frac{\Delta Q}{\epsilon}$, μ is a horizontal deviation, b is a scale variable and x is a variable.

Definition 4. *(Post-Processing). Given a randomized algorithm* A *that satisfies* ϵ*-differential privacy,* F *is an arbitrary randomized function. Then a randomized algorithm* F$\cdot$ A *satisfies* ϵ*-differential privacy.*

Definition 5. *(Wavelet transformationation). As a special technique in mathematics,* DWT*(discrete wavelet transformation) can divide an input discrete sample into* AC*(approximation coefficients) and* DC*(detail coefficients),which respectively correspond to the low frequency and high frequency parts of the original sample. Such a wavelet decomposition process can be carried out recursively up to the expected decomposition. On the contrary,* IDWT *(inverse discrete wavelet transformation) can recombine* AC *and* DC *into the original sample. The* AC *and* DC *are respectively defined as follows:*

$$AC = \sum_{k=-\infty}^{\infty} x(\mathrm{k})g(2l-k)$$

$$DC = \sum_{k=-\infty}^{\infty} x(\mathrm{k})h(2l-k).$$

where g *is a low frequency filter and* h *is a high frequency filter. In AC,* ac_{jl} *denotes the* j-th *approximation coefficients in the* l-th *level of decomposition.*

In this paper, we choose a well known Haar wavelets, which has the simplest orthogonal basis among all discrete wavelet transforms. The scaling function of Haar wavelet transform is represented by S, which is indicated as follows:

$$s = \begin{cases} 1 & 0 \leq x < 1 \\ 0 & otherwise \end{cases}$$

The mother wavelet of Haar wavelet transform is denoted by M, which is described as follows:

$$M(x) = \begin{cases} 1 & 0 \leq x < 0.5 \\ -1 & 0.5 \leq x < 1 \\ 0 & otherwise \end{cases}$$

Definition 6. *(degree centrality). The degree centrality of node* v_i *is the sum of the number of adjacent nodes, which is denoted by* Cd(v_i)*. Formally, the degree centrality is given by:*

$$Cd(v_i) = \sum_{j=1}^{n} a(v_i, v_j)$$

where a(v_i,v_j*) denotes the edge between node* v_i *and node* v_j*. In general, a(*v_i,v_j*) equals 1.*

In a weighted graph, the weight degree centrality of node v_i is the sum of the weights of edges which connect node v_i. We can define the weight degree centrality as:

$$Cd_w(v_i) = \sum_{j=1}^{n} w(v_i, v_j)$$

where w(v_i,v_j) represents the weight of edge between node v_i and node v_j.

Definition 7. *(The between centrality). The between centrality of node* v_i *is given by:*

$$\mathrm{Cb}(v_i) = \sum_{j,k} \frac{p(v_j, v_i, v_k)}{p(v_j, v_k)}$$

where $p(v_j, v_k)$ *denotes the number of shortest paths between node* v_j *and node* v_k*,* $p(v_j, v_i, v_k)$ *is the number of shortest paths between node* v_j *and node* v_k *which go* *through v_i. In a weighted graph, the betweenness centrality is defined by:*

$$\mathrm{Cb}_w(v_i) = \sum_{j,k} \frac{p_w(v_j, v_i, v_k)}{p_w(v_j, v_k)}$$

where $p_w(v_j, v_k)$ is the sum of edge weight in shortest paths between node v_j and node v_k, $p_w(v_j, v_i, v_k)$ is the sum of edge weight in shortest paths between node v_j and node v_k which go through v_i.

4 DPEW Method

In this section, we propose a DPEW method to preserve edge weight privacy in a social weighted network when it is published. In this method, we devise two algorithms: WTDP algorithm and EDU algorithm. WTDP algorithm can achieve differential privacy preserving while adding less noise on the edge weights, and EDU algorithm can maintain the properties of original weighted social network. In addition, we prove that DPEW method satisfies differential privacy while obtaining better data utility.

4.1 The Model of DPEW Method

For preserving edge weight privacy in the weight social networks, we introduce a practical method that combines wavelet transform with differential privacy, which also maintains the shortest path length between some important nodes in original weight social network unchanged. The frame structure of proposed method is illustrated in Figure 1.

In this model, the input is an original weighted social network, which has sensitive information: the edge weights. The output is a published weighted social network which is preserved by differential privacy. In order to provide rigorous privacy guarantee, wavelet transformation and differential privacy are combined in WTDP algorithm, which satisfies ϵ-differential privacy. Owing to the deficiency of data utility caused by the Laplace noise, this model present EDU algorithm which aims to preserve the character of original weighted social network for enhancing data utility. Therefore, our method can not only preserve the privacy of the original weighted social network but can also keep the data utility of the published weighted social network.

4.2 WTDP and EDU Algorithm

4.2.1 WTDP Algorithm

With the application of wavelet transformation, we propose a new algorithm which adds less noise to achieve differential privacy for weights of edges. In this algorithm, we first get a weight sequence of edge weights W and apply wavelet transformation on it, After that, we gain the approximation coefficients and the detail coefficients of sequences wavelet transformation. According to Laplace mechanism, we add Laplace noise to the approximation coefficients to achieve differential privacy. Thus, we can generate a preserved weighted graph by using inverse wavelet transformation. The frame structure of WTDP algorithm is illustrated in Figure 2 and Algorithm 1.

Algorithm 1 The WTDP algorithm

```
Input: The original weighted social network: $G$=($V$, $E$, $W$);the best decomposition level: $C$ ; privacy budget: $\epsilon$;
Output: The noised weighted social network: $G'$=($V$, $E$, $W'$)
1: $Wm \leftarrow Max(W)$
2: Sensitivity: $\Delta$ f = $\frac{Wm}{C}$
3: $b \leftarrow \frac{\Delta f}{\epsilon}$
4: $Wa \leftarrow$ wavelet transform in $W$
5: for $wa_i$ in $Wa$:
6:     A Laplace noise $n_i \leftarrow$ Laplace(b)
7:     Adding $n_i$ on $wa_i$
8: $Wa' \leftarrow Wa$
9: $W' \leftarrow$ Inverse Wavelet transformation $Wa'$ and $Wd$
10: Return noised weighted social network: $G'$=($V$, $E$, $W'$)
```

In WTDP Algorithm, we input a social weighted network G, the best decomposition level C and privacy budget ϵ. For preserving edge weights in this social weighted network G, we first get the max edge weight in line 1. Then, from line 2 to line 3 the scale parameter b in Laplace distribution is obtained. Line 4 describes the wavelet transformation of W and gains approximation coefficients Wa. In line 5-8, for every wa_i in Wa, a Laplace noise n_i is added and we get the perturbed approximation coefficients Wa'. By using the perturbed approximation coefficients Wa' and the original detail coefficients, line 9 describes the inverse wavelet transformation and obtains a noised edge weight sequence W'. Finally, we get an edge weight sequence W' which is preserved by WTDP algorithm.

4.2.2 EDU Algorithm

For minimizing the changes of edge weight and achieving better data utility, we propose an algorithm to keep some characters of original social weighted network unchanged in the noised social weighted network. In order to achieve this purpose, we select some important nodes which possess large degree centrality and betweenness centrality in original social weighted network and make the shortest distance between these selected important nodes in the noised social weighted network equal to that in the social weighted network graph. Thus, the description of EDU algorithm is given in Figure 3 and Algorithm 2.

In this algorithm, we use the composite parameters Nc to evaluate the importance of node, which is shown as follows.

$$Nc = \sqrt{Cd_w(v_i)^2 + Cb_w(v_i)^2}$$

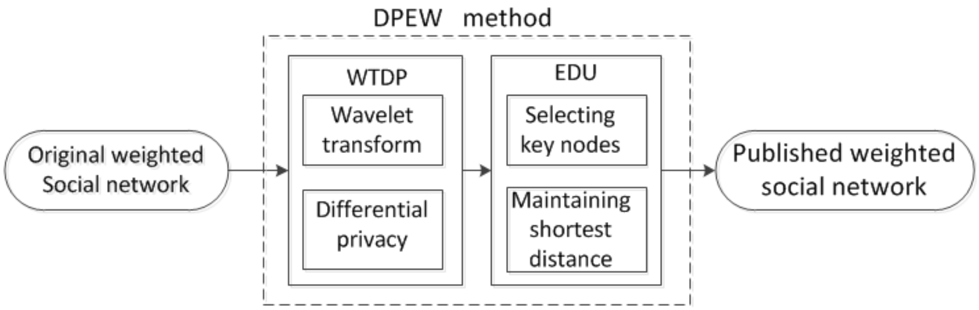
DPEW method
WTDP
Wavelet transform
Differential privacy
EDU
Selecting key nodes
Maintaining shortest distance
Original weighted Social network
Published weighted social network

Figure 1: The model of DPEW method

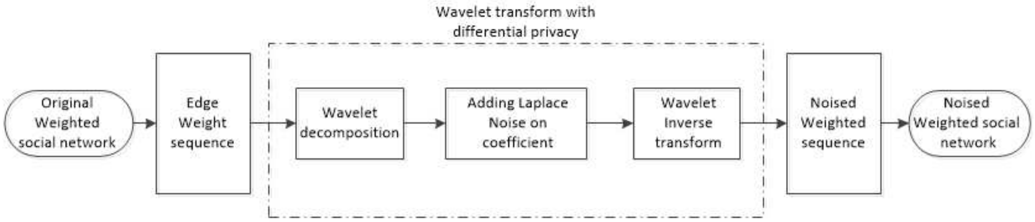
Wavelet transform with differential privacy
Original Weighted social network
Edge Weight sequence
Wavelet decomposition
Adding Laplace Noise on coefficient
Wavelet Inverse transform
Noised Weighted sequence
Noised Weighted social network

Figure 2: The WTDP algorithm

where $Cd_w(v_i)$ denotes the weight degree centrality of v_i, $Cb_w(v_i)$ represents the betweenness centrality of v_i. The larger Nc, the more important node v_i is. When the shortest distance between two important nodes is maintained unchanged in a noised social weighted network, the shortest distance between any two nodes in a noised social weighted network will be closed to that in the original social weighted network. If more shortest distance are kept unchanged in noised weight social network, there may be less perturbation on the published weight social network.

For the EDU algorithm, the detail is described as follows. Firstly, we select k important nodes according to the value of Nc, then we work out the shortest distance among those nodes in the original social weighted network and in the noised social weighted network. Secondly, we make the edge weights in the noised social weighted network to be equal to these in the original social weighted network. At last, we get a perturbed social weighted network which not only preserves the original social weighted network but also gains better data utility.

In EDU algorithm, we calculate Nc of nodes in an original weight social network in line 1. Line 3 generates the important node sequence D' after selecting k nodes according to value of Nc. Line 4 to line 9 outline how to keep the shortest distance of nodes in D' unchanged in G''. Line 4 and line 7 calculate the shortest path of nodes in the original weight social network and get the edges list L_e and edge weights list W_e in shortest path, The modification of the edge weight in G^* is described in line 8 and line 9, which keep the length of the shortest path in G unchanged. Finally, this algorithm returns a perturbed weight social network G'' which preserves the shortest distance length in the original weight social network.

Algorithm 2 The EDU algorithm

```
Input: weighted social network: $G$=($V$, $E$, $W$); noised weight social network: $G'$=($V$, $E$, $W'$)
Output: perturbed weight social network: $G''$=($V$, $E$,$W''$)
1: Calculating $Nc$ of nodes in $G$
2: Selecting $k$ nodes from $V$ according to value of $Nc$
3: Generating a sequence $D'$ containing $k$ important nodes
4: for $i$ in $D'$:
5:    for $j$ in $D'$:
6:        Calculate shortest path from node $i$ to node $j$ in $G$
7:        Get edges list $L_e$ and edge weights list $W_e$ in shortest path
8:        Keep the length of shortest path unchanged in $G'$
9:        Modifying the edge weight in $G'$
10:Return perturbed weighted social network: $G''$=($V$, $E$,$W''$)
```

4.3 Theoretical Analysis

Given a weighted social network G=(V, E, W), W is the edge weight sequence. After transforming the W into wavelet domain, we get Wa, which denotes the approximation coefficients, and Wd, which represents the detail coefficients.

Assume that two weighted social networks, G_1 and G_2 are neighbors, and the difference between G_1 and G_2 is two edges. Let $Q(\bullet)$ be a query function Q :$G\rightarrow$ Wa, so $Q(G_1)$=Wa_1, $Q(G_2)$=Wa_2. According to the definition

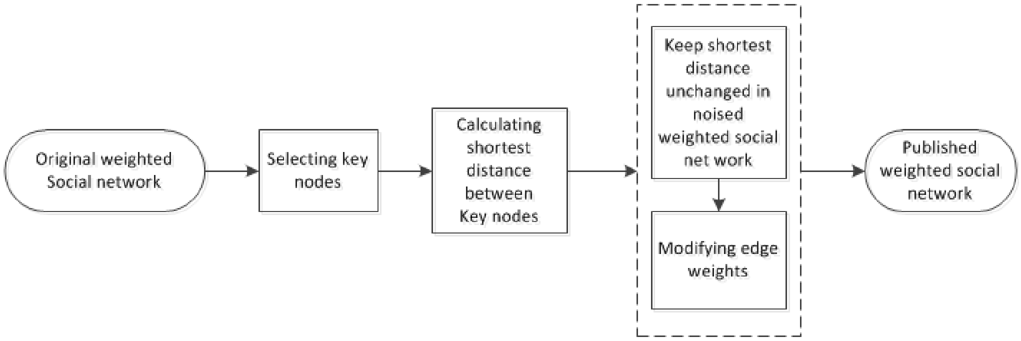
Original weighted Social network
Selecting key nodes
Calculating shortest distance between Key nodes
Keep shortest distance unchanged in noised weighted social net work
Modifying edge weights
Published weighted social network

Figure 3: The EDU algorithm

of sensitive, we get the sensitivity of Q:

$$\Delta Q = max_{G_1,G_2}|Q(G_1) - Q(G_2)|_1$$

$$\begin{aligned}\Delta Q &= max|Wa_1 - Wa_2|_1 \\ &= max|(wa_{11}, wa_{12}, ..., wa_{1m}) - (wa_{21}, wa_{22}, ..., wa_{2m})| \\ &= \frac{\Delta(w_{max} - w_{min})}{2^{ND}}\end{aligned}$$

where w_{max} and w_{min} are the maximum and minimum values in the W, ND is the level of decomposition. Then, we add the Laplace noise to the output of Q in accordance with the Laplace Mechanism, where LA is the Laplace Mechanism.

Let Pr $[G_1]$ denotes the probability density function of LA (G_1, Q, ε), and Pr $[G_2]$ indicates the probability density function of $LA(G_2$, Q, ε). Then, the proof is described as follows.

$$\begin{aligned}&\frac{Pr[LA(G_1)]}{Pr[LA(G_2)]} = \frac{Pr[\eta(G_1)]}{Pr[\eta(G_2)]} \\ &= \frac{Pr[R - Q(G_1)]}{Pr[R - Q(G_2)]} \\ &= \frac{\frac{1}{2\frac{\Delta Q}{\epsilon}} exp(-\frac{|R-Q(G_1)|}{\frac{\Delta Q}{\varepsilon}})}{\frac{1}{2\frac{\Delta Q}{\epsilon}} exp(-\frac{|R-Q(G_2)|}{\frac{\Delta Q}{\varepsilon}})} \\ &= \frac{exp(-\frac{|R-Q(G_1)|}{\frac{\Delta Q}{\varepsilon}})}{exp(-\frac{|R-Q(G_2)|}{\frac{\Delta Q}{\varepsilon}})} \\ &= exp(\frac{\varepsilon|R - Q(G_1)|}{\Delta Q} - \frac{\varepsilon|R - Q(G_2)|}{\Delta Q}) \\ &= exp(\frac{\epsilon(|R - Q(G_1)| - |R - Q(G_2)|)}{\Delta Q}) \\ &\leq exp(\frac{\epsilon(|Q(G_1) - Q(G_2)|)}{\Delta Q}) \\ &\leq exp(\frac{\epsilon\Delta Q}{\Delta Q}) = e^{\epsilon}\end{aligned}$$

Therefore, we can achieve differential privacy preserving for Wa. After conducting IDWT on the noised approximation coefficients Wa and detail coefficients, we generate a noised edge weight sequence $W^{'}$, which can be used to construct a noise weight social network. For better data utility, we carry out modifying the noised $W^{'}$ while maintaining the properties of original weighted social network. At last, we construct a published weighted social network based on the $W^{''}$ to preserve the original weighted social network. According to the requirement of post-processing, we achieve differential privacy preserving for original weighted social network with better data utility.

5 Experiments and Results

5.1 Datasets

In experiments, there are two kinds of data sets: the synthetic weighted network data and the real weighted network data. All the data sets used are shown below.

1) Synthetic weighted network. In the experiment with synthetic data, we generate two random graphs with 500 nodes and 1,000 nodes, which are randomly connected to each other with probability p=0.3. For each edge, an integer weight is assigned randomly in the range [1,200]. We call this synthetic graph as Random Graph.

2) Real weighted network. In the Windsurfers network, there are 43 nodes and 336 edges, which contains interpersonal contacts between windsurfers in southern California during the fall of 1986. The Infectious SocioPatterns dataset contains the daily cumulated networks represented in the Infectious SocioPatterns visualization, which includes 307 nodes and 1924 edges. The weights associated with the edges are the number of 20 seconds intervals during which close-range face-to-face proximity has been detected. The high-energy theory collaborations (Hetc) data set is a weighted network of coauthorships between scientists posting preprints on the High-Energy Theory E-Print Archive between Jan 1, 1995 and December 31, 1999. It has 5835 nodes and 13815 edges.

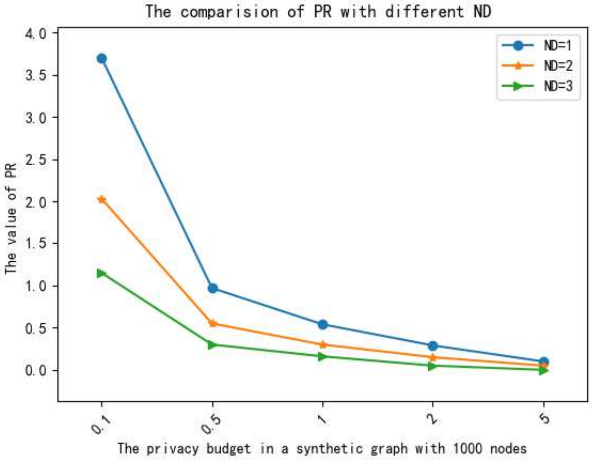

Figure 4: Comparison of different *ND* in a synthetic graph

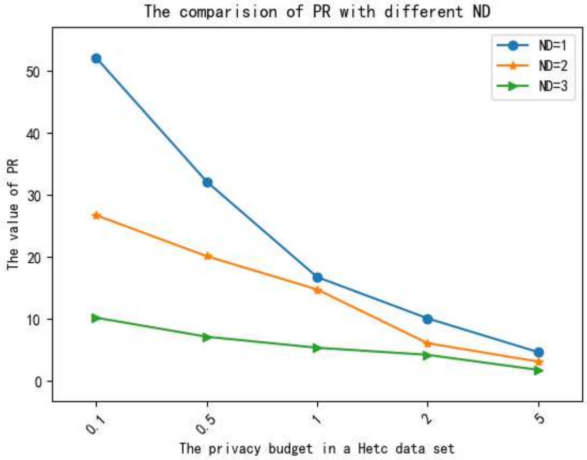

Figure 5: Comparison of different *ND* in a Hetc data set

5.2 Privacy Evaluation

For evaluating our method, we take advantage of the PR (perturbation ratio) to measure the performance in privacy preserving. Moreover, we compare our method with other methods in privacy preserving.

5.2.1 Privacy Measurement

In this section, the perturbation ratio (*PR*) is used to measure the performance of preserving privacy, which is the ratio of *(Wp -W)* to *W*, where the perturbed edge weight is *Wp* and the original edge weight is *W*. The larger *PR*, the better privacy preserving.

$$PR = \frac{W_p - W}{W} = \frac{\sum_{i=1}^{n} |wp_i - w_i|}{\sum_{i=1}^{n} w_i}$$

Meanwhile, *ND*, the number of wavelet decomposition, which is equal to the level of decomposition *l*, can determine the sensitivity in our method. If we want better privacy preserving, we can decrease *ND*, otherwise, we can increase *ND* for better data utility.

In order to compare with our method, we select four methods: GR method(Gaussian randomization method) [10], k-anonymization mothod [15], Edge-DP method(edge-differential privacy based method), DP-MB method(differential privacy based on merger of barrels method) [9]. In the experiments, we set ϵ in [0.1, 0.5, 1, 2, 5], the *ND* is set in [1,2, 3]. Due to the uncertainty of the noise, we execute all data sets 10 times by using our approach and other approaches to average out the results.

5.2.2 Privacy Analysis

In privacy analysis, we first conduct the experiment on the synthetic data sets and real data sets by using our method and keep the experiment results in Table 1. As shown in Table 1, when *ND* is 1, ϵ is 5, the *PR* in synthetic graphs with 500 nodes and 1,000 nodes is respectively 0.11 and 0.10, while the *PR* in three real data sets is 0.63, 0.10, 0.10 respectively. If we decrease ϵ from 5 to 0.1, the *PR* in synthetic graphs with 500 nodes increases from 0.11 to 3.72, as does the PR in other data sets. This result indicates that the smaller ϵ, the better privacy preserving. When ϵ is 2, if we increase *ND* from 1 to 3, the *PR* in synthetic graphs with 500 will decrease from 0.30 to 0.05, as will the PR in other data sets, which shows that *ND* can affect the privacy preserving.

Next, we describe the changing tendency of *PR* in our method with ϵ varying in Figure 4 and Figure 5, where *ND* is from 1 to 3, respectively. As shown in Figure 4, when ϵ increases from 0.1 to 5, *PR* in synthetic graphs with 1000 nodes decreases simultaneously no matter how much *ND* is. When ϵ is a fixed value, the value of *PR* declines as the value of *ND* increases, which means the wavelet transformation can control the privacy preserving of the method. In Figure 5, the *PR* in a Hetc data set is same as that in Figure 4. By using *PR*, it is clear that our method can achieve privacy preserving for the edges. In addition, for better understanding the comparison among different methods in a synthetic graph with 1000 nodes and a Hetc data set, the details are demonstrated in Figure 6 and Figure 7 respectively. When ϵ is from 0.1 to 5, *ND* is 2, the *PR* obtained by these methods in a synthetic graph with 1000 nodes is illustrated in Figure 6, where the *PR* in our method is larger than DP-MB method and smaller than that in the other three methods. Specially, the change of *PR* in GR method is small when ϵ increases from 0.5 to 5. In a Hetc data set, *PR* in our method, which is described in Figure 7, is smallest in those five methods no matter what ϵ is. All the results show that our method can improve data utility owing to adding less noise to edge weights.

To sum up, the experimental results show that our method can achieve differential privacy preserving for weighted graphs. In addition, by using the wavelet transform in our method, we can control the Laplace noise

Table 1: The value of PR in our method

ND	ϵ	**synthetic data1**	**synthetic data2**	**Windsurfers network**	**SocioPatterns**	**Hetc**
1	0.1	3.72	3.70	21.66	40.15	52.11
1	0.5	0.97	0.97	5.82	11.74	32.16
1	1	0.55	0.54	3.06	8.15	16.78
1	2	0.30	0.29	1.06	6.26	10.12
1	5	0.11	0.10	0.63	5.05	4.69
2	0.1	2.05	2.03	9.14	16.72	26.73
2	0.5	0.56	0.55	2.79	6.95	20.14
2	1	0.29	0.30	1.30	5.69	14.78
2	2	0.16	0.15	0.78	4.99	6.12
2	5	0.05	0.05	0.37	4.63	3.16
3	0.1	1.18	1.15	4.86	9.11	10.22
3	0.5	0.30	0.30	1.24	5.17	7.16
3	1	0.16	0.16	0.74	4.72	5.38
3	2	0.05	0.05	0.34	4.48	4.46
3	5	1.5e-16	1.5e-16	0.18	5.24	2.18

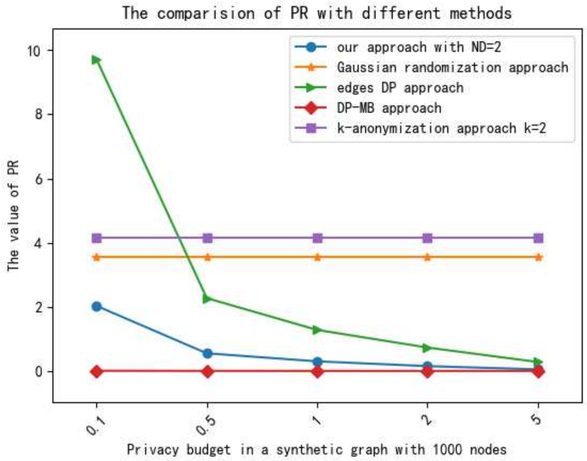

Figure 6: Comparison of different method in a synthetic graph(*ND*=2)

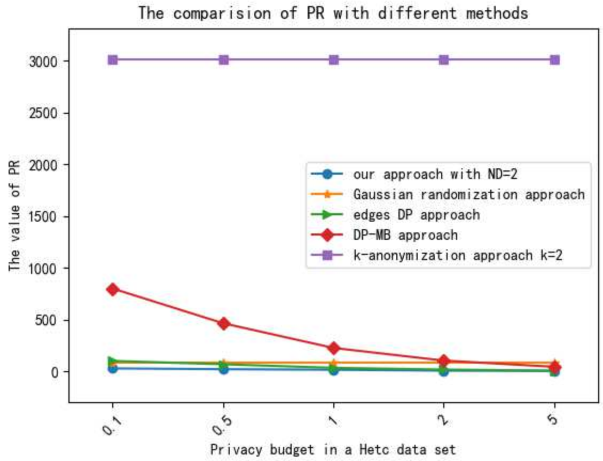

Figure 7: Comparison of different method in a Hetc data set(*ND*=2)

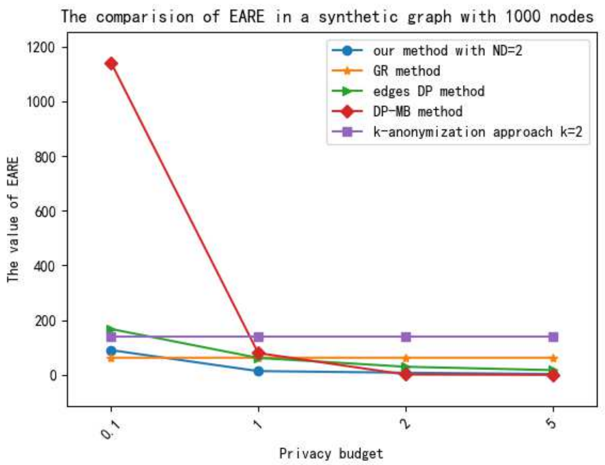

Figure 8: Comparison of *EARE* in a synthetic graph

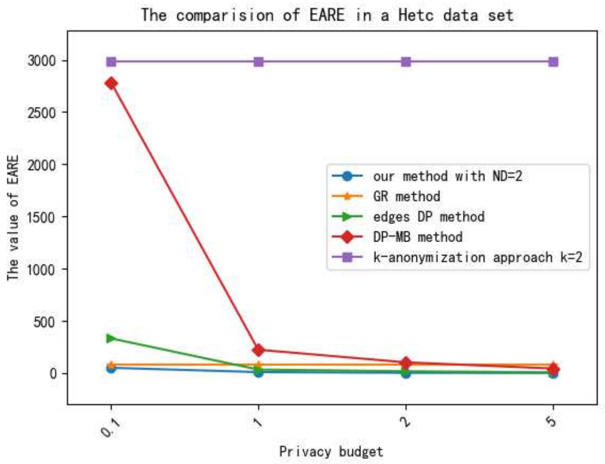

Figure 9: Comparison of *EARE* in a Hetc data set

which is added on the edge weights. Therefore, our method gains better privacy preserving than DP-MB method and has better data utility than GR method, k-anonymization mothod, and edge-DP method.

5.3 Utility Evaluation

In this section, we define some metrics of the graph to evaluate the data utility. Then, after analyzing and discussing our method in data utility, we compare our method with other methods.

5.3.1 Utility Metrics

To evaluate the data utility, we use four metrics: *EARE* (edge average relative error), *NARE* (node average relative error), *ASD* (average shortest distance) and *KSPL* (Keeping Shortest Path length).

1) *EARE*. *EARE* is the average relative error of edge weight, which indicates the edge change caused by privacy preserving. The smaller the value, the higher data utility.

$$EARE = \frac{\sum_{i=1}^{n} |Wp_i - W_i|}{n}$$

where Wp_i denotes the edge weight in published weighted social network, W_i represents the edge weight in original weighted social network.

2) *NARE*. *NARE* is the average relative error of node weight, which describes the node change caused by perturbation. The smaller *NARE*, the better data utility.

$$NARE = \frac{\sum_{i=1}^{n} |VWp_i - VW_i|}{n}$$

where VWp_i denotes the node weight in published weighted social network, VW_i represents the node weight in original weighted social network.

3) *ASD*. *ASD* is an important property of the weighted graph, which is the average shortest distance among all pairs of nodes.

$$ASD = \sum_{s,t \in V} \frac{d(s,t)}{n(n-1)}$$

where V is the set of nodes in G, $d(s,t)$ is the shortest path from s to t, and n is the number of nodes in G.

4) *KSPL*. *KSPL* is the proportion of unchanged shortest path length.

$$KSPL = \frac{Np^{'}}{Np}$$

where Np^* is the number of unchanged shortest path lengthen in in published weighted social network, while Np denotes the total number of shortest path length in original weighted social network. The larger *KSPL*, the more the shortest path lengths are unchanged.

5.3.2 Utility Analysis

In this experiment, we set ϵ in [0.1,1,2, 5] and *ND* in 2. In addition, four methods, such as GR method (Gaussian randomization method) [10], k-anonymization mothod [15], Edge-DP method(edge-differential privacy based method), DP-MB method(differential privacy based on merger of barrels method), are used for comparison. Due to the uncertainty of the noise, we conduct our method and other methods 10 times to average out the results.

In the utility analysis, first of all, we discuss the experimental results gained by our method. As shown in Table 2, when ϵ is 0.1, *ND* is 2, the results of *EARE*, *NARE*, *ASD*, *KSPL* in a synthetic data set with 500 nodes

Table 2: Utility metrics in our method ND=2

data sets	***metrics***	**original network**	ϵ**=0.1**	ϵ**=1**	ϵ**=2**	ϵ**=5**
synthetic data1	*EARE*	0	90.30	13.73	7.79	2.90
synthetic data1	*NARE*	0	8072.21	553.83	202.96	53.32
synthetic data1	*ASD*	6.12	4.66	4.20	4.56	5.09
synthetic data1	*KSPL*	1	0.08	0.12	0.16	0.19
synthetic data2	*EARE*	0	88.92	14.13	8.23	3.03
synthetic data2	*NARE*	0	15990.36	1197.27	460.88	147.30
synthetic data2	*ASD*	4.57	4.13	3.12	3.23	3.59
synthetic data2	*KSPL*	1	0.06	0.12	0.14	0.16
Windsurfers network	*EARE*	0	76.88	15.58	10.92	5.37
Windsurfers network	*NARE*	0	1195.11	226.58	158.77	75.79
Windsurfers network	*ASD*	2.19	12.25	4.95	3.77	2.53
Windsurfers network	*KSPL*	1	0.065	0.066	0.065	0.047
SocioPatterns	*EARE*	0	124.05	41.60	36.24	32.40
SocioPatterns	*NARE*	0	1526.51	493.70	429.52	384.53
SocioPatterns	*ASD*	4.66	71.17	23.33	16.26	11.26
SocioPatterns	*KSPL*	1	0.05	0.08	0.10	0.11
Hetc	*EARE*	0	221.12	21.16	10.12	4.32
Hetc	*NARE*	0	1081.24	112.99	58.94	18.17
Hetc	*ASD*	4.57	253.78	72.45	28.74	13.22
Hetc	*KSPL*	1	0.04	0.05	0.06	0.08

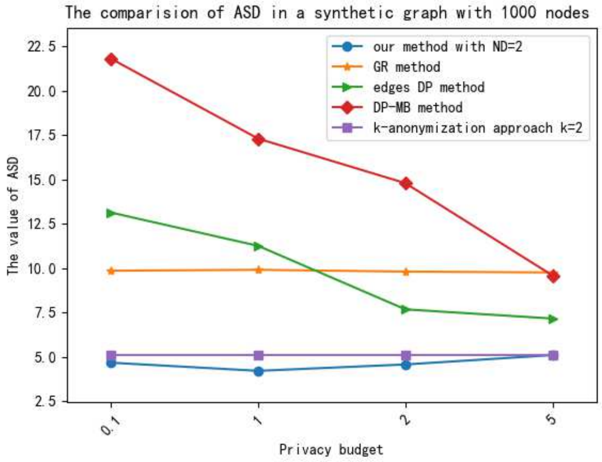

Figure 10: Comparison of ASD in a synthetic graph

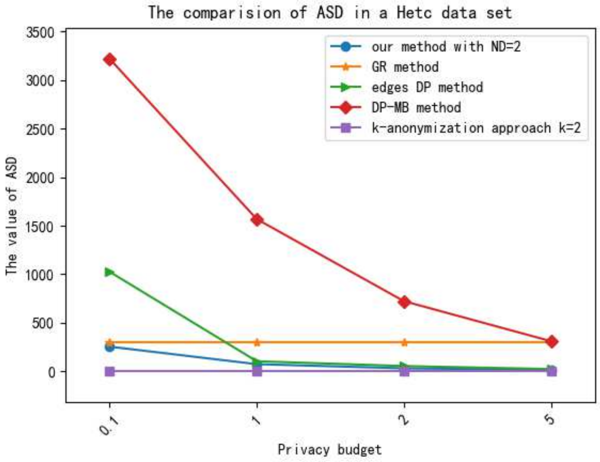

Figure 11: Comparison of ASD in a Hetc data set

are 90.30, 8072.21, 4.66, 0.08, respectively. When ϵ is increased to 5, the values of $EARE$ and $NARE$ will decrease simultaneously together. In addition, the value of ASD is close to that in the original graph because ASD is mostly associated with the number of the selected important nodes. In particular, the value of $KSPL$ changes slightly. Furthermore, it is worth noting that the results in other data sets are equivalent to those in the synthetic data set with 500 nodes. All the results state clearly that data utility will be improved with the increase of ϵ.

Next, particularly when ϵ is changed from 0.1 to 5 in a synthetic data set with 1000 nodes and a Hetc data set, the comparison of these methods is illustrated by these figures as follows. As shown in Figure 8 and Figure 9, the values of $EARE$ in different methods decline with ϵ increasing. Specially, in Figure 8, the value of $EARE$ in our method is smaller than that in the k-anonymization mothod, the edge-DP method, and the Edge-DP method when ϵ increases from 0.1 to 5, while it is larger than that in the GR method as ϵ is less than about 0.5. In Figure 9, we can see that the value of $EARE$ in our method is smallest in these mothods. As illustrated in the Figure 10, the change of the ASD in different methods and the values of ASD in other four methods are larger than that in our method. For example, when ϵ equals to 1, the values in other four methods are 5.21,10.01, 11.24,16.45, respectively, while the value in our method is 4.2. In addition, when ϵ is smaller than 1, the value of ASD obtained by our method is the larger than that in k-anonymization mothod, which is shown in the Figure 11. Therefore, the result shows that our method can obtain a better data utility compared with other methods.

Finally, owing to the wavelet transform and post-processing, the results indicate that our method can achieve better performance in data utility than GR method, k-anonymization mothod, DP-MB method and edge-DP method. Therefore, we can see that our method can improve the data utility while satisfying the differential privacy.

6 Conclusions

For preserving the privacy data of social networks, the differential privacy which is able to provide strict privacy guarantee has been extensively applied. Compared with other differential privacy based methods, in this work, we focus on achieving differential privacy for edge weights while keeping the data utility as much as possible and publishing a preserved weighted social network. Therefore, we propose a method which combines wavelet transform with differential privacy. In this method, we first apply the wavelet transform on the edge-weight sequence and add the Laplace noise to the wavelet coefficients, then we take advantage of inverse wavelet transform to realize differential privacy. At last, for modifying the error of shortest distance of noised graph, a special algorithm is used to improve the data utility. In addition, we present two algorithms: WTDP algorithm and EDU algorithm. To evaluate the performance of our method, the PR is used to evaluate the privacy preserving of different methods when ϵ is fixed. Moreover, the theory analysis and experimental results show that our method not only satisfies ϵ-differential privacy but also improves data utility. In the future, due to the perturbation caused by the stochastic noise in ϵ-differential privacy, we must work hard to maintain the property of graph while satisfying ϵ-differential privacy.

Acknowledgments

This study was supported by the National Natural Science Foundation of China under grant No. 61962033, the Natural Science Basic Research of Shaanxi province under grant No. 2020JM-288, the Fundamental Research Funds for the Central Universities under grant No. GK201704017 and No. GK201903011.We are gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] W. Y. Day, N. Li and M. Lyu, "Publishing graph degree distribution with node differential privacy," in *Proceedings of 10th Theory of Cryptography Conference*, pp. 133-138, 2016.

[2] M. R. E. Dishabi, M. A. Azgomi, "Differential privacy preserving clustering using Daubechies-2 wavelet transformation," *International Journal of Wavelets, Multiresolution and Information Processing*, vol. 13, no. 14, pp. 1550028, 2015.

[3] C. Dwork, "Differential privacy," in *Proceedings of the 33rd International Colloquiium on Automata, Languages and Programming*, pp. 1-12, 2006.

[4] S. Guo, M. Wen, X. H. Liang, "A differentially private k-means clustering scheme for smart grid," *International Journal of Network Security*, vol. 23, no. 1, pp. 126-134, 2021.

[5] M. Hay, C. Li, G. Miklau, *et al.*, "Accurate estimation of the degree distribution of private networks," in *Proceedings of 9th IEEE International Conference on Data Mining*, pp. 169-178, 2009.

[6] M. Hay, G. Miklau, D. Jensen, *et al.*, "Anonymizing social networks," in *Computer Science*, pp. 07-19, 2007. (`https://scholarworks.umass.edu/cgi/viewcontent.cgi?referer=https://www.google.com/&httpsredir=1&article=1175;context=cs_faculty_pubs`)

[7] S. P. Kasiviswanathan, K. Nissim, S. Raskhodnikova, A. Smith, "Analyzing graphs with node-differential privacy," in *Proceedings of 10th Theory of Cryptography Conference*, pp. 457-476, 2013.

[8] J. Leskovec, C. Faloutsos, "Scalable modeling of real graphs using kronecker multiplication," in *Proceedings of the 24th International Conference on Machine Learning*, pp. 497-504, June 2007.

[9] X. Y. Li, J. Yang, Z. L. Sun, ”Differential privacy for edge weights in social networks,” *Security and Communication Networks*, vol. 2017, pp. 1-10, 2017.

[10] L. Liu, J. Wang, J. Liu, *et al.*, ”Privacy preserving in social networks against sensitive edge disclosure,” *Preserving Data Privacy in Knowledge Discovery*, 2008. (`https://www.researchgate.net/publication/228972647_Privacy_preserving_in_social_networks_against_sensitive_edge_disclosure`)

[11] L. Liu, J. Wang, J. Zhang, ”Wavelet-based data perturbation for simultaneous privacy-preserving and statistics-preserving,” in *Proceedings of International Conference on Data Mining Workshops*, pp. 27-35, 2008.

[12] T. Ma, Y. Zhang, J. Cao, J. Shen, M. Tang, Y. Tian, A. Al-Dhelaan, M. Al-Rodhaan, ”KDVEM: A k-degree anonymity with vertex and edge modification algorithm,” *Computing*, vol. 97, no. 12, pp. 1165-1184, 2015.

[13] D. Mir, R. N. Wright, ”A differentially private estimator for the stochastic kronecker graph model,” in *Proceedings of Joint EDBT/ICDT Workshops*, pp. 167-176, 2012.

[14] F. Nagle, L. Singh, A. Gkoulalas-Divanis, ”EWNI: Efficient anonymization of vulnerable individuals in social networks,” in *Proceedings of the 16th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, pp.359-370, 2012.

[15] M. E. Skarkala, M. Maragoudakis, S. Gritzalis, ”Privacy preservation by k-anonymization of weighted social networks,” in *Proceedings of International Conference on Advances in Social Networks Analysis and Mining*, pp. 423-428, 2012.

[16] C. Task, C. Clifton, ”What should we protect? Defining differential privacy for social network analysis,” *State of the Art Applications of Social Network Analysis*, pp. 139-161, 2014.

[17] Y. C. Tsai, S. L. Wang, T. P. Hong, ”Extending [K1, K2] anonymization of shortest paths for social networks,” in *Proceedings of International Conference on Multidisciplinary Social Networks Research*, pp. 187-199, 2015.

[18] X. Xiao, G. Wang, J. Gehrke, ”Differential privacy via wavelet transformations,” *IEEE Transactions on Knowledge & Data Engineering*, vol. 23, no. 8, pp. 1200-1214, 2011.

[19] X. Ying, X. Wu, ”On link privacy in randomizing social networks,” in *Proceedings of Pacic-Asia Conference on Advances in Knowledge Discovery and Data Mining*, pp. 28-39, 2009.

[20] X. Ying, X. Wu, ”Randomizing social networks: A spectrum preserving approach,” in *Proceedings of the SIAM International Conference on Data Mining*, pp.739-750,2008.

[21] J. Zhang, G. Cormode, C. M. Procopiuc, *et al.*, ”Private release of graph statistics using ladder functions,” in *Proceedings of ACM SIGMOD International Conference on Management of Data*, pp. 731-745, 2015.

[22] Y. B. Zhang, Q. Y. Zhang, Y. Yan, *et al.*, ”A k-Anonymous location privacy protection method of polygon based on density distribution,” *International Journal of Network Security*, vol. 23, no. 1, pp. 57-66, 2021.

Biography

Jun Yan received the M.S. degree in College of Earth Exploration Science and Technology, Jilin University(2007). He is currently pursuing the Ph.D. degree in College of Computer Sciensce, Shaanxi Normal University. His research interests include network security and privacy preserving.

Hai Liu received his B. S. degree (2012) and M.S. degree (2015) from Guizhou University,and obtained Ph.D. degree (2019) from School of Computer Science, Shaanxi Normal University. His main research interest includes privacy protection.

Zhenqiang Wu received his B.S. degree in 1991 from Shaanxi Normal University, China, and received his M.S. and Ph.D degrees in 2002, and 2007 respectively, all from Xidian University, China. He is currently a full professor of Shaanxi Normal University, China. Dr. Wu's research interests include computer communications networks, mainly wireless networks, network security, anonymous communication, and privacy protection etc. He is a member of ACM and senior of CCF.