A graph generating method based on local differential privacy for preserving link relationships of social networks

Jun Yan[1], Yijun Zhang[2], Laifeng Lu[2], Yi Tian[3], and Yihui Zhou[4]
[1]School of Mathematics and Computer Applications, Shangluo College, Shangluo, Shaanxi, 72600, China
[2]School of Mathematics and Statistics, Shaanxi Normal University, Xiâan, Shaanxi, 710119, China
[3]School of Economics and Management, Shangluo College, Shangluo, Shaanxi, 72600, China
[4]School of Computer Science, Shaanxi Normal University, Xiâan, Shaanxi, 710119, China

With the widespread popularity of social networks, there are serious privacy issues related to the graph data of social networks. To address these issues, many differential privacy based graph generating methods have been proposed. However, these methods mainly focus on preserving the properties of the graph and ignore the preservation of node link relationships. To further preserve the link relationship of each user in the distributed environment of Social Networks while providing effective data utility, a local differential privacy graph generating method is proposed, in which the randomized response is utilized to modify the link relationships of nodes and all modified subgraphs of each node are merged to get a synthetic graph to preserve the link privacy of each node. In addition, the 2-hop subgraph based node encoding of each node is adopted to reduce the disturbance caused by the local differential privacy. The unbiased estimate of random response and the node similarity are applied to maintain data utility. Theoretical analysis demonstrates that the designed method satisfies differential privacy while maintaining data utility. The experimental results indicate the effectiveness of this method.

***Index Terms*—local differential privacy, link relationship, node coding, randomized response, unbiased estimate**

I. Introduction

With the rapid development of network technology, various types of Internet have emerged, such as the Internet of Vehicles[1], the Internet of Things[2] and Satellite Internet[3]. Based on the underlying support of various types of Internet, social networks have developed from a single social platform to a social meta universe[4]. In addition, social network-based application platforms utilize digital technology to provide services for people's lives and the entire society, forming a digital social network application environment. Therefore, social networks have played an increasingly important role in people's lives and the entire society.

However, while social networks are widely used, they also face enormous problems and challenges. In recent years, the issue of data privacy has received increasing attention[5]. As a platform for data applications, social networks can collect a large amount of personal data. However, these personal data contain sensitive information such as name, gender, date of birth, education level, marital status, etc. If these sensitive personal information are not properly protected, it can lead to the leakage of personal privacy[6]. For example, Facebook's Cambridge scandal erupted in 2018, resulting in the leakage of the personal information of 80 million users, which raised public concerns about personal privacy[7]. TikTok was fined $5.7 million by the Federal Trade Commission in February 2019 for violating the Children's Online Privacy Protection Act while collecting user video data[8]. These events indicate serious privacy issues in social networks. Therefore, effective measures must be taken to resolve these issues in time.

In particular, social networks contain not only tabular data but also graph data that represents the characteristics of social networks[9]. Graph data consists of nodes and links, where nodes represent users and links represent social relationships between users[10]. On the one hand, the nodes of graph data carry a large amount of sensitive personal information of users. On the other hand, the topological structure of graph data represents the structure of social networks in a meaningful way, containing many network attributes information such as degree distribution, shortest distance, and clustering coefficient[11]. The privacy risks of graph data in social networks mainly include three aspects[12], namely identity leakage, membership relationship leakage, and content leakage.

To preserve the privacy of graph data in social networks, many methods based on perturbation have been proposed, such as graph modification and differential privacy. Compared with the graph modification methods, differential privacy has significant advantages and is therefore widely used for preserving graph structured data. Usually, differential privacy mainly preserves the parameters and statistical values of graph data during querying and publishing[13], including degree distribution, count queries, and degree histograms. In addition, differential privacy is also applied to generate a synthetic graph, thereby preserving the information of the original graph[14]. However, it should be noted that the scenario for these differential privacy applications is that all data is stored in the hands of trusted data collectors, who implement privacy preservation. Therefore, differential privacy in this scenario is called centralized differential privacy. However, in reality, it is often difficult to find reliable data collectors. If data collectors misuse user information, it can lead to personal privacy breaches. For example, Facebook's unauthorized sale of insights and analysis based on users' personal data has raised public concerns about personal privacy issues[15].

In response to the shortcomings of central differential privacy, local differential privacy is proposed to enhance the

Manuscript received October 22, 2024; revised January 13 2025. Corresponding author: Laifeng Lu (email: lulaifeng@snnu.edu.cn).

strength of data privacy preservation. For the graph data, local differential privacy mainly preserves the parameters and statistical values of graph data[16] or generates synthetic graphs[17]. Compared to preserving only specific graph data metrics, generating synthetic graphs provides a more general paradigm that can effectively achieve data sharing and analysis. To generate synthetic graphs, Qin et al.[18] presented the LDPGen method. In [19], Wei et al. proposed the AsgLDP method to generate attribute synthesis graphs. In addition, Hou et al.[20] designed the first dynamic graph publishing method based on local differential privacy: the PPDU method. However, these local differential privacy methods only focus on preserving the properties of the graph, which results in resulting in insufficient preservation of node link relationships. To preserve the link privacy of each node in the original graph by local differential privacy, the following three key issues need to be addressed. Firstly, how to achieve privacy preservation for the original graph through perturbations of each node is important. Secondly, because the perturbations of nodes bring a large number of redundant edges to the original graph, how to reduce the perturbations of redundant edges is a difficult problem. Finally, it is necessary to improve the data utility of the noised graph obtained through perturbations.

In this work, a local differential privacy graph generating method has been proposed to address the issues above. In this method, local differential privacy is applied to enhance the preservation of node link relationships. In addition, three measures are presented to maintain the data utility. First, a perturbation method based on node 2-hop subgraphs is proposed. Secondly, in the 2-hop subgraph encoding sequence, the frequency distribution of value 1 corresponds to the frequency distribution of edges in the subgraph. To reduce the number of redundant edges, a method of edge frequency correction is adopted for the 2-hop subgraph encoding sequence with added noise to improve the accuracy of the number of edges. Finally, it is necessary to consider the structural connections between nodes when generating a node subgraph, which is beneficial to merge subgraphs to obtain a synthetic graph that is as similar as possible to the original graph.

Our contributions of this paper are summarized as follows:

(1)We propose a local differential privacy graph generating method, which can preserve the link privacy of each node while maintaining data utility.

(2)We devise a node coding algorithm based on the 2-hop subgraph of each node and make use of the randomized response to modify the edges of each node. In addition, the properties of unbiased estimate and the similarity between two nodes are utilized to maintain data utility.

(3)We employ several kinds of real datasets to evaluate the designed method in term of privacy preservation and data utility, and the experiment results show that the method is effective in practice.

The organization of this paper is as follows:

The graph modification methods and differential privacy based methods are introduced in section 2. In section 3, Some related definitions are described. Section 4 mainly shows how an LDPGG method preserves the link privacy of graph data, and the details and theoretical analysis of all algorithms are illustrated in Section 5. Section 6 demonstrates the performance of the LDPGG method in term of privacy preservation and data utility. Finally, section 7 presents the conclusion and the future work.

II. RELATEDWORK

To preserve the graph structure data in social networks, many graph modification methods were first proposed, such as edge and node modification methods, generalization methods, and uncertain graph methods. After that, many differential privacy based methods were also developed to provide stronger privacy preservation.

In the existing edge and node modification methods, X.Ying[21] presented two algorithms for preserving the graph data while maintaining the spectral properties of original graph unchanged as much as possible. In addition, the *k*-anonymity method was adopted to constrain the perturbation caused by edge modification[22]. Casas.R[23] used exhaustive search and greedy algorithms to obtain anonymous degree sequences which is similar to the original graph, and then designed neighbor centrality and random edge selection methods. Finally, edge modifications were minimized to get a anonymous graph. Mortazavi.R[24] developed a (*k*, *l*) graph modification method based on *k* with edge addition, which achieved the desired trade-off between data utility and privacy preservation. Considering attacks from both graph structures and vertex attributes, Ren in [25] presented a novel mechanism to preserve graph privacy, in which the original graph was divided into a so-called *kt*-safe graph, via *k*-anonymity and *t*-closeness. Moreover, to preserve the large scale graph data, [26] devised an anonymity framework, in which a *k*-anonymity algorithm based on *k*-decomposition was presented.

In generalization methods, to minimize the structural information loss in the generalization process, [27] used the GA and PSO to devise several hybrid solutions form the super-nodes of size at least *k*. In addition, a graph clustering method based on structure entropy in [28] was proposed to preserve graph data in SIoT, in which data mining was combined with structural information theory.

In uncertain graph methods, Boldi first utilized the concept of uncertainty to design a (*k*, ¦*Å1*)-obfuscation method, which could generate an uncertain graph to preserve the original graph[29]. In addition, Mittal in [30] proposed a Rand-Walk method, which was able to provide stronger privacy preserving than the (*k*, ¦*Å1*)-obfuscation method. Based on the work mentioned above, Nguyen devised a generalized obfuscation model, in which the uncertain adjacency matrices was adopted to obtain an uncertain graph while keeping the degree of nodes unchanged[31]. To resist link relationship attacks based on background knowledge, [32] proposed an edge-differential privacy based uncertain graph method, which further improved the privacy preservation capability of the uncertain graph method.

As differential privacy could prevent any attacks based on background knowledge and provide rigorous mathematical proof[33], many differential privacy methods had been proposed to preserve graph data. In many methods based on central differential privacy, there were usually two applications:

preserving specific sensitive statistics of graphs and publishing private graphs. In [34], utilizing a heuristic truncation strategy and a new privacy budget allocation strategy, Xing designed a DP-gSpan method that mined frequent subgraphs under differential privacy. When publishing the triangle counts under node-differential privacy, [35] used a novel graph projection method to get an upper bound for sensitivity in differential privacy. Therefore, the designed mechanism attained better accuracy for the triangle counts while satisfying differential privacy.To release the degree histogram under node-DP, a method was presented in [36], in which the mean filtering and some techniques were introduced to further improve publishing accuracy. To publish graphs under node-DP, [37] devised two methods. One perturbed the original graph by randomly inserting and removing nodes, while the other one modified the input graph by randomly adding nodes and removing edges. To prevent degree attacks while preserving graph structure, [38] proposed a privacy preservation approach called PBCN, which combined clustering and randomization algorithms. In [39], V.Karwa used differential privacy to obtain a graphical degree partition of a graph and used it to generate a synthetic graph. Moreover,the *DK*-1 sequence and *DK*-3 sequence were preserved by differential privacy to gain a synthetic graph, which could also preserve the original graph[40].

Particularly, due to its stronger privacy protection capabilities than central differential privacy, local differential privacy was widely applied to preserve graph data. In [41], to publish graph statistics including k-stars, triangles, and 4-cycles, a one-round algorithm was designed to count k-stars by using an optimal order. By far, another one-round algorithm based on random response was proposed to preserve triangles. Furthermore, a LDP-enabled graph metric estimation framework for graph analysis was designed in [16]. In this framework, a complete or parameterized algorithm was proposed to simplify jobs in implementing LDP-related steps, and the privacy budget between the two atomic was optimally allocated. To generate a synthetic graph, Qin presented the LDPGen method in [18], which was a novel multi-phase technique. In this method, after the information from each user was injected with noise according to LDP and reported to a collector, all users with similar structures were clustered by using an optimal parameter. Then, a social graph generation model was employed to preserve the original social graph. In [19], AsgLDP was developed to preserve an attribute graph. To improve data utility, various graph properties were maintained by carefully injecting noise. Moreover, the LDP based method for dynamic graph publication was designed in[20]. In this approach, a privacy-preference-specifying mechanism was adopted to reduce noise injection.

III. Preliminaries

In general, a graph *G*= (*V*, *E*) represents a social network, where *V* is a node set in it and *E* represents link relationships of nodes.

Definition 1 (Differential Privacy).

Given $\varepsilon \geq 0$, for any two neighboring datasets D_1 and D_2,that they differ in one record, and all *S* that is in the output of a algorithm *Z*, the following holds:

$$P_r[Z(D_1) \in S] \leq exp(\varepsilon) P_r[Z(D_2) \in S] \quad (1)$$

where ε is a privacy budget.Thus, the algorithm *Z* satisfies ε-differential privacy. For graph data, there are edge differential privacy and node differential privacy.

Definition 2 (Local Differential Privacy).

Given $\varepsilon \geq 0$, for any two inputs *t* and $t^{'}$ (t ,$t^{'} \subseteq$ *Dom* (*Z*)), if the probability that the algorithm *Z* gets the same output result t^*($t^* \subseteq$ *Ran*(*Z*)) is

$$P_r[Z(t) = t^*] \leq exp(\varepsilon) P_r[Z(t^{'}) = t^*] \quad (2)$$

where *Ran*(*Z*) and *Dom* (*Z*) are the input and output domains of the algorithm *Z*, a randomized algorithm *Z* satisfies ε-differential privacy.

If there is one different edge between two inputs t and $t^{'}$, the algorithm *Z* satisfies edge local differential privacy. If two inputs t and $t^{'}$ differ by one node, the algorithm *Z* satisfies node local differential privacy.

To achieve local differential privacy, the random response mechanism is usually used. In addition, local differential privacy has post-processing and Parallel composition properties.

Definition 3 (Randomized Response). The define of randomized response is shown as follows:

$$P(y_i = k | x_i = j) = P_{jk} \quad (3)$$

where x_i represents an input that is *j*, P_{jk} denotes the probability to get an output y_i which is *k*.

When *j* and *k* is in {0,1},the 2-dimensional randomized response is as follows:

$$P_m = \begin{pmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{pmatrix} \quad (4)$$

where P_m is the design matrix, in which the sum of probabilities of each row equals 1.

Let $\{\pi_1,\ \pi_2\}$ denote the proportions of respondents$^{'}$ true values which fall in each of the input values in {0,1}. Let λ_1, λ_2 be the empirical probabilities of the observed values.

Thus, there is $(\lambda_1,\ \lambda_2)^T$= $P^T(\pi_1,\ \pi_2)^T$. In addition, an unbiased estimator $\hat{\pi}$ is $\hat{\pi}$ = $(P^T)^{-1}\hat{\lambda}$, where $\hat{\lambda}$=(λ_1, λ_2)represents the vector of observed empirical probabilities.

Definition 4(Randomized Response satisfying ε-Differential Privacy).

Let $\varepsilon \geq 0$, if max P_{00}/P_{10}, P_{00}/P_{01}, P_{01}/P_{11}, P_{10}/P_{11} $\leq$ e^{ε} , the randomized response represented by P_m satisfies ε-differential privacy.

Definition 5(Post-Processing).

For any one data set D,if a randomized algorithm Z satisfies ε-differential privacy, D can be preserved by the algorithm Z and $D^{'}$ is also obtained, which denotes the result of the algorithm Z.

If N is an arbitrary randomized algorithm, when $D^{"}$ is gained by using N on $D^{'}$, the algorithm *Z*·*N*: $D \rightarrow D^{"}$ satisfies ε-differential privacy.

Definition 6 (Parallel composition properties).

For n algorithms Z_1, Z_2, ..., Z_n, in which each algorithm Z_i satisfies ε_i-differential privacy, if n disjoint subsets of the input database D is preserved by each algorithms Z_i respectively, this process provides *Max* ε_i-differential privacy for the data set D, which is the parallel composition properties of differential privacy.

IV. Model of method

A. Motivation

To preserve the link relationships of social networks, many graph modification methods have been presented. However, this kind of method can not resist attacks based on back knowledge. Compared with the graph modification method, differential privacy methods can provide stronger privacy preservation. Typically, there are two types of differential privacy: central differential privacy and local differential privacy. Especially, local differential privacy focuses on perturbing each user in social networks during the data collection process and possesses stronger privacy preservation than central differential privacy. In addition, the graph generation method provides a universal paradigm that can preserve graph data during the process of data collection and data release.

Thus, motivated by this, a local differential privacy graph generation method is utilized to preserve the link relationships of each user in a social network. In particular, the randomized response is applied to preserve the link relationships of each user. Additionally, the node encoding based on the 2-hop subgraph, the property of unbiased estimate and the similarity between two nodes are utilized to maintain data utility.

Ultimately, the proposed method can preserve the link relationships of the original social network while providing effective data utility.

B. Framework

Based on the above research ideas, a local differential privacy graph generating method is designed. In Fig.1, the entire framework possesses two parts: the client end and the data collection end.

On the client side, in order to control the disturbance range of nodes, the 2-hop subgraph of each node is selected as the disturbance object. Then, edge local differential privacy is applied for perturbing the nodes. In the specific implementation, the 2-hop subgraphs of each node are first encoded, and then the randomized response is utilized for perturbation. Ultimately, all users will send the perturbed data to the data collection end.

On the data collection end, the data receiver synthesizes the data of all users to obtain a noisy graph, and optimizes each user's data received to reconstruct the subgraph of each node. After the edge frequency of each user's data received is estimated, the edge count of each node is optimized based on the estimated value, and then the similarity between two nodes and the 2-hop subgraph of each node are calculated based on the noised graph. In accordance with the similarity size, each node selects a series of nodes and adds edges between them. The selected number of nodes is the optimized number of node edges. Then each node generates a reconstructed 2-hop subgraph, and all reconstructed 2-hop subgraphs are merged to gain a synthetic graph, which can provide privacy preservation for each node.

C. Work Process

The specific work process of the local differential privacy graph generating method is demonstrated in Fig.2, which mainly consists of three steps.

In step 1, each user extracts their own link relationships to get their own 2-hop subgraph, and then obtains the encoding sequence through the subgraph encoding. Then, Step 2 takes advantage of the randomized response to perturb the encoding sequence and send all noised sequences to an untrusted data collector. In step 3, an unbiased estimate of the frequency of the number of edges for each node is gained, which can be used to obtain the optimal number of edges for each node. Then, based on the similarity between nodes, each node selects nodes from its 2-hop subgraph to link, resulting in a synthetic subgraph. In the end, all synthetic subgraphs are merged to generate a synthetic graph. Especially, due to the fact that the disturbance revolves around a 2-hop subgraph of nodes, the entire process effectively maintains the structure of the original graph.

Therefore, the local differential privacy graph generation method effectively protects the link privacy of each node and provides effective data utility.

V. algorithms and analysis

Based on the presented model, the *LDPGG*(local differential privacy graph generating) algorithm is designed. In the *LDPGG* algorithm, three algorithms are proposed: *NEHG*(Node Encoding based on 2-Hop Sub-Graph) algorithm, *NSRR*(Node Sequence Randomized Response) algorithm, and *SGG*(Synthetic Graph Generation) algorithm.

A. Algorithms

1) LDPGG algorithm

The entire algorithm process is described as follows:

```
Algorithm 1  LDPGG algorithm
Input: an original graph $G$
Output: a synthetic graph $G_s$
1: a node set $V \leftarrow$ an original graph $G$
2: a set of node encoding sequence $S_E \leftarrow$ NEHG algorithm(V)
3: a set of noised node encoding sequence $S_{En} \leftarrow$ NSRR algorithm($S_E$)
4: a synthetic graph $G_s \leftarrow$ SGG algorithm($S_{En}$)
5: Return a synthetic graph $G_s$
```

First of all, assuming that each node obtains its 2-hop subgraph through communication between its friends, each node achieves its node encoding through the *NEHG* algorithm. Then, after disturbing each node's 2-hop subgraph encoding sequence, the *NSRR* algorithm preserves the link relationships of each node. Finally, the *SGG* algorithm obtains the synthetic graph by merging all synthetic subgraphs of nodes.

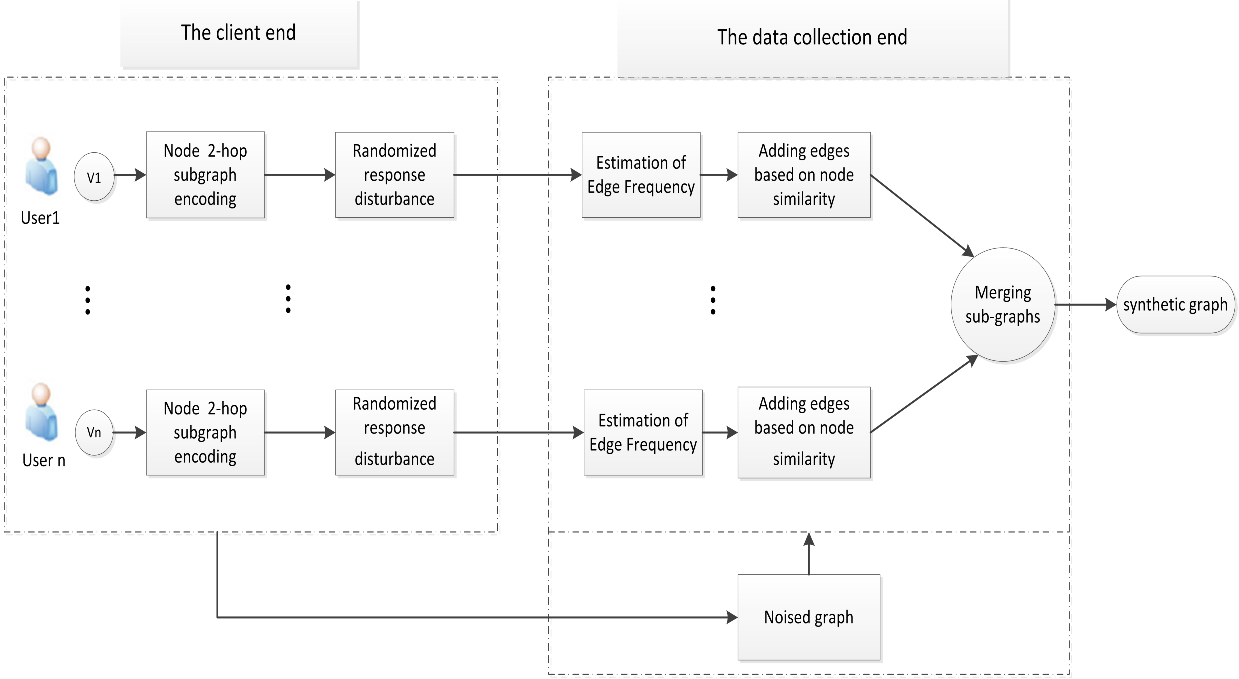
The client end
The data collection end
User1
V1
Node 2-hop subgraph encoding
Randomized response disturbance
Estimation of Edge Frequency
Adding edges based on node similarity
User n
Vn
Node 2-hop subgraph encoding
Randomized response disturbance
Estimation of Edge Frequency
Adding edges based on node similarity
Merging sub-graphs
synthetic graph
Noised graph

Fig. 1: The framework of the proposed method

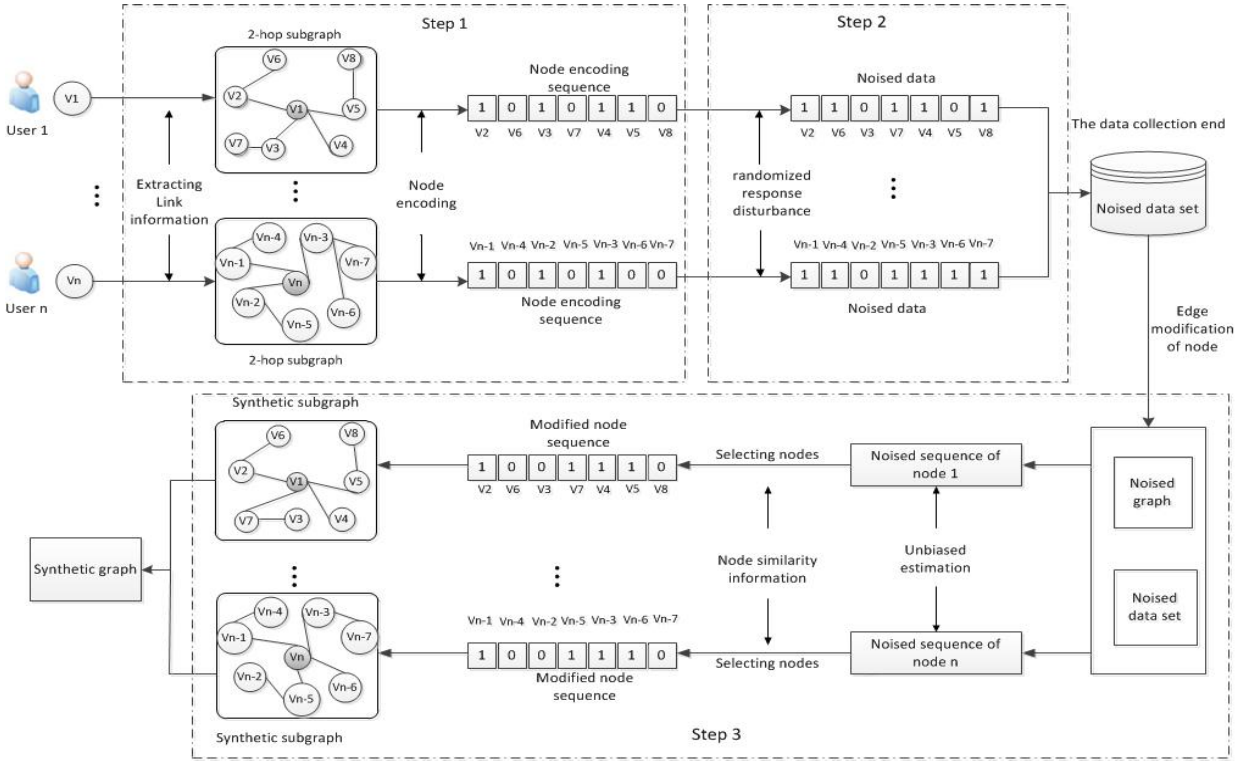

Fig. 2: The process of the proposed method

Among these algorithms, to reduce disturbance, the frequency estimation property of random response is utilized for edge frequency estimation, thereby obtaining an approximate value of the number of edges. In addition, the similarity between nodes is used to construct node subgraphs that are similar to the original node subgraphs.

2) NEHG algorithm

To preserve the link relationships of each node, the Randomized Response is applied to disturb all nodes. Since the Randomized Response usually deals with 0 or 1, 0 and 1 are used to encode link relationships of nodes. In addition, to reduce the disturbance caused by the Randomized Response, the 2-hop subgraph of each node is exploited to get a node encoding sequence.

In the *NEHG* algorithm, each node is encodeed from line 2 to line 11. Firstly, each node v_i utilizes its link relationships to generate its own 2-hop subgraph S_{Gvi}, through which a node set S_{vi} is obtained in line 3. Then, in accordance with the link situation between node v_i and node v_j, each node v_i can be encodeed and a sequence of node encoding $Sn_{vi}$is obtained from line 5 to line 10. In particular, 1 is used to record that node v_i and node v_j is connected while 0 denotes that node v_i does not link node v_j. Therefore, a [0,1] sequence Sn_{vi} describes the link relationships of node v_i and represents the node coding of node v_i. After all node coding sequences are added into a set node v_i in line 11, finally, line 12 return a set of node coding sequence S_{ec}.

Algorithm 2 *NEHG* algorithm

```
Input: a node set $V$
Output: a set of node encoding sequence $S_{ec}$
1: a set of node encoding sequence $S_{ec}$ = {}
2: for $v_i$ in $V$
3:     2-hop subgraph $S_{Gvi} \leftarrow$ extracting link information of $v_i$
4:     a node set $S_{vi} \leftarrow S_{Gvi}$
5:     a node sequence $Sn_{vi}$ with | $S_{vi}$ | zeros $\leftarrow S_{vi}$
6:     for $v_j$ in $S_{vi}$
7:         if $v_i$ connecting $v_j$
8:             1 $\leftarrow$ the value of $v_j$ in $Sn_{vi}$
9:         else
10:            0 $\leftarrow$ the value of $v_j$ in $Sn_{vi}$
11:        a set of node encoding sequence $S_{ec}$ adding $Sn_{vi}$
12: Return a set of node encoding sequence $S_{ec}$
```

3) NSRR algorithm

In the *NSRR* algorithm, to preserve the link privacy of each node, the randomized response is used to perturb the node encoding sequence of each node v_i. At last, this algorithm return a set of noised node encoding sequence S_{en}, which achieves the privacy preservation for the each node. Line 3 obtains a node encoding sequence S_{evi} of the node v_i from a set of node encoding sequence S_{ec}. Then, the randomized response perturbs a node encoding sequence S_{evi} and a noised encoding sequence S_{envi} is generated in line 4. After that, line 6 adds S_{envi} to the set S_{ecn}. Finally, a set of noised node encoding sequenceS_{ecn} is obtained in line 6.

Algorithm 3 *NSRR* algorithm

```
Input: a set of node encoding sequence $S_{ec}$
Output: a set of noised node encoding sequence$S_{ecn}$
1: a set of noised node encoding sequence $S_{ecn}$ = {}
2: for i in $S_{ec}$
3:     a node encoding sequence $S_{evi} \leftarrow S_{ec}$[i]
4:     $S_{envi} \leftarrow$ RR disturbance($S_{evi}$)
5:     $S_{ecn}$ addes $S_{envi}$
6: Return a set of noised node encoding sequence $S_{ecn}$
```

4) SGG algorithm

To reduce the disturbance brought by local differential privacy, the unbiased estimate of randomized response and the similarity between nodes are adopted to maintain data utility. For a noised node encoding sequence S_{cnvi}, the unbiased estimate of the frequency distribution of number 1 in this sequence is got through the unbiased estimate of randomized response. Then, the approximate value of the number of edges that the node links *m* can be obtained. After that, in each noised subgraph of each node, after calculating the node similarity between nodes, each node selects *m* nodes to link according to the value of the node similarity. Afterwards, a modified node encoding sequence is obtained and each noised subgraph of each node is modified. Finally, all modified node subgraphs are merged to generate a synthetic graph, which can preserve the link relationships of each node in the original graph.

Algorithm 4 *SGG* algorithm

```
Input: a set of noised node encoding sequence $S_{ecn}$
Output: a synthetic graph $G_s$
1: a noised graph $G_c \leftarrow S_{ecn}$
2: a set of modified node encoding sequence $S_{ecm}$ = { }
3: for $i$ in $S_{ecn}$
4:     a noised node encoding sequence $S_{envi}$ , a node $v_i \leftarrow$ emphS$_{ecn}$[i]
5:     the number of 1 in $S_{envi} \leftarrow$ Unbiased estimation $S_{envi}$
6:     $m \leftarrow$ the number of 1 in $S_{envi}$
7:     a node set $S_{vi} \leftarrow S_{envi}$
8:     for j in $S_{vi}$
9:         calculating the node similarity between nodes $v_i$ and $v_j$ in $S_{vi}$
10:    selecting $m$ nodes according to the similarity
11:    $S_{emvi} \leftarrow$ correcting $S_{envi}$
12:    a set of modified node encoding sequence$S_{em}$ adds $S_{emvi}$
13: for $n$ in $S_{em}$
14:    a modified subgraph $Gm_{sub} \leftarrow$ generating graph $S_{em}$[n]
15: a synthetic graph $G_s \leftarrow$ merging all $Gm_{sub}$
16: Return a synthetic graph $G_s$
```

In the *SGG* Algorithm, the noise graph G_c is firstly got through a set of noised node encoding sequence S_{ecn}. Then, from line 3 to line 12, each noised node encoding sequence in S_{Ecn} is corrected to get a set of modified node encoding sequence S_{ecm}. After the noised node encoding sequence S_{envi} of node v_i is obtained in line 4, line 5 gains the frequency estimation of value 1 in the sequence through unbiased estimation. Based on the frequency estimation of value 1, the approximate value *m* of the number of edges of node v_i is got in line 6. Afterwards, node v_i calculates the node similarity between itself and the nodes in S_{envi}, and selects *m* similar nodes from high to low according to the value of the node similarity. Then, line 11 corrects the S_{envi} and a set of modified node encoding sequenceS_{em} is gained in line 12. For each node, a modified subgraph Gm_{sub} of each node is generated through the modified node encoding sequence S_{em}[n]. Finally, line 15 merges all modified subgraph Gm_{sub} to gain a synthetic graph G_s.

B. Analysis

To demonstrate the effectiveness of the *LDPGG* algorithm, the algorithm analysis is presented. Firstly, it is proven that the *LDPGG* algorithm satisfies differential privacy, which indicates its privacy preservation capability. Then, the analysis of data utility shows that the *LDPGG* algorithm can provide effective data utility. Finally, the computational complexity of the *LDPGG* algorithm is analyzed.

1) Privacy analysis

Theorem 1: The *NSRR* algorithm satisfies differential privacy.

Proof: Let *R* be a randomized response mechanism, $P_r[x \rightarrow y]$ represents the probability that $x \in \{0,1\}$ changes to $y \in \{0,1\}$.

Given q=1-p= $e^{\varepsilon}/1+e^{\varepsilon}$, when $\varepsilon \geq 0$, $q \geq p$. For each node, a binary sequence based on the 2-hop subgraph of each node can be obtained. Let two sequences $Se_1(e_1,e_2,...,e_n)$ and $Se_2(e_1{}',e_1{}',...,e_n{}')$ be neighbor sequences in which there is one different element between them.

Without loss of generality, let $M(m_1,m_2,...,m_n)$ be any output of *R*. Assuming e_1 is not equal to $e_1{}'$, there is the following result.

$$\begin{aligned}
&\frac{P_r[R(Se_1)=M]}{P_r[R(Se_2)=M]} \\
&= \frac{P_r[e_1 \rightarrow m_1]\cdot \ldots \cdot P_r[e_n \rightarrow m_n]}{P_r[e_1{}' \rightarrow m_1]\cdot \ldots \cdot P_r[e_n{}' \rightarrow m_n]} \\
&= \frac{P_r[e_1 \rightarrow m_1]}{P_r[e_1{}' \rightarrow m_1]} \leq \frac{q}{p} = e^{\varepsilon}
\end{aligned}$$

Therefore, regardless of whether the input is Se_1 or Se_2, the *NSRR* algorithm satisfies differential privacy.

Theorem 2: The *LDPGG* algorithm satisfies differential privacy.

Proof: In this algorithm, each node is preserved by the randomized response mechanism, achieving differential privacy preservation for the node. According to parallel composition property of local differential privacy, the process of protecting all nodes in a graph satisfies differential privacy. Moreover, in accordance with the post-processing rule, the process of generating a synthetic graph by using a set of the noised node encoding sequence also satisfies differential privacy. Therefore, the *LDPGG* algorithm satisfies differential privacy.

2) Utility analysis

In the encoding sequence of the 2-hop subgraph of node n_i, assuming there are *n* nodes v_1, v_2,...,v_n and each node v_i has a binary value $x_i \in \Omega 1$, where $\Omega 1$ is a set {0,1}. In addition, this value is considered as the sensitive attribute *X* of each node v_i. 1 indicates that node n_i links node v_i , while 0 shows that node n_i is not connected to node v_i. When the randomized response perturbs the encoding sequence of the 2-hop subgraph of node n_i, y_i is a random value of the true value x_i of each node v_i, where $y_i \in \Omega 2$ and $\Omega 2$ is {0,1}.

Let π represent the true proportion of sensitivity value 1 in the set of subgraph nodes. The proportion of the observed value "1" in the data collected after randomized response is λ. In addition, $\hat{\pi}$ denotes an unbiased estimate of π.

Theorem 3: Given the transformation matrix of the random response mechanism

$$P = \begin{pmatrix} p & 1-p \\ 1-q & q \end{pmatrix}$$

, an unbiased estimate of π is

$$\hat{\pi} = \frac{\lambda-(1-p)}{p+q-1}$$

.

Proof:

Let *n* represent the total number of nodes in the 2-hop subgraph encoding sequence of node n_i, and n_1 be the total number of nodes with attribute "1". If y_i denotes the response obtained from the *i*-th node, then

$$\begin{aligned}
P_r(y_i=1) &= q\pi + (1-p)(1-\pi) \\
P_r(y_i=0) &= (1-q)\pi + p(1-\pi)
\end{aligned}$$

Then the maximum likelihood function can be obtained as follows:

$$L = [\pi q + (1-\pi)(1-p)]^{n_1}[\pi(1-q)+(1-\pi)p]^{n-n_1}$$

From the above equation, it can be concluded that:

$$\begin{aligned}
lnL = n_1 ln[(p+q-1)\pi + (1-p)] \\
+(n-n_1)ln[(1-p-q)\pi + p]
\end{aligned}$$

$$0 = \frac{(p+q-1)n_1}{(p+q-1)\pi+(1-p)} + \frac{(n-n_1)(1-p-q)}{(1-p-q)\pi+p}$$

Simplifying the above equation yields the following result:

$$\hat{\pi} = \frac{n_1/n-(1-p)}{p+q-1} = \frac{\lambda-(1-p)}{p+q-1}$$

Next, it will be proven that the above estimates are unbiased.

In fact, it is easy to know that y_i follows a Bernoulli distribution, so the expectation of y_i is

$$E(y_i) = q\pi + (1-p)(1-\pi)$$

Then

$$\begin{aligned}E(\hat{\pi}) &= E(\frac{n_1/n-(1-p)}{p+q-1})\\ &= \frac{E(\sum_{i=1}^{n} y_i)/n-(1-p)}{p+q-1}\\ &= \frac{\sum_{i=1}^{n} E(y_i)/n-(1-p)}{p+q-1}\\ &= \frac{nE(y_i)/n-(1-p)}{p+q-1}\\ &= \frac{[\pi q+(1-\pi)(1-p)]-(1-p)}{p+q-1}\\ &= \pi\end{aligned}$$

Therefore, the estimated expected value of $\hat{\pi}$ is $E(\hat{\pi})$=π, which is the unbiased estimate of π.

By replacing the true proportion π of sensitive value 1 in the subgraph node set, an approximate value of the number of nodes with attribute 1 in the node n_i sequence can be obtained, as well as an approximate value of the number of edges of node n_i.

In summary, after the 2-hop subgraph sequence of each node n_i is perturbed by the randomized response, to minimize the disturbance on each node as much as possible, the unbiased estimate is used to get an approximate value of the number of edges of node n_i. In addition, after obtaining the number of node edges, node n_i make use of the similarity between the two nodes to select nodes for connection, so that the generated synthetic graph can maintain the structure the original graph as much as possible. Therefore, the data utility of the *LDPGG* algorithm can be guaranteed.

3) Algorithm complexity analysis

Assuming there are n nodes, each node v_i has k_i neighboring nodes in its 2-hop subgraph, and the maximum value of k_i in all nodes is k_{max}, which is much smaller than n. The *NESG* algorithm encodes each node with a complexity of $O(nk_{max})$. The *NRR* algorithm implements perturbations on each node, with an algorithm complexity of $O(n\mathrm{k}_{max})$. The *GSG* algorithm merges all sub-graphs to obtain a synthetic graph, with an algorithm complexity of $O(n\mathrm{k}_{max}+n)$. Therefore, the complexity of the *LDPGG* algorithm is $O(nk_{max})$+$O(nk_{max})$+$O(n\mathrm{k}_{max}+n)$, which ultimately can be reduced to $O(nk_{max})$.

VI. Experiments

In this section, the experimental condition is first introduced. Then, the performance of *LDPGG* algorithm is demonstrated in terms of privacy preservation. Finally, the data utility evaluation of the *LDPGG* algorithm is described in detail.

A. Experimental condition

In experiments, four real datasets are used to evaluate the *LDPGG* algorithm. Blogs data sets records front-page hyperlinks between blogs with 1,224 nodes and 19025 edges. There are two types of Face-book data sets. One has 4039 nodes and another has 63731 nodes.Enron email network has 36692 nodes.

In order to evaluate the effectiveness of the *LDPGG* algorithm, the *LDPGG* algorithm is compared with the graph modification based on *k*-anonymity algorithm[23] and the *LDPGen* algorithm[18]. To reduce the randomness caused by disturbances, all data sets are executed 10 times by all methods to get the average value.

All experiments are run on a HP computer with a 5.00GHz Intel Core i7-8500 and 32GB memory. In addition, Python is applied in programm on the Microsoft Windows 7 operating system.

B. Privacy evaluation

1) Privacy measurement

In general, the edit distance between two graphs describes the similarity between two graphs. To evaluate the *LDPGG* algorithm, Euclidean Distance between two graphs is used to measure the effectiveness of privacy preservation.Euclidean Distance between two graphs is as follows:

$$\begin{aligned}ED(G,G_s) &= ED(V,V_s)+ED(E,E_s)\\ &= \sqrt{\sum_{i=1}^{n}(degree_{v_i}-degree_{v_{si}})^2+\sum_{i=1}^{n}(e_i-e_{si})^2}\end{aligned}$$

Where the first part of *ED* is the change of node degrees, The latter part describes the number of different edges.

Clearly, the larger the *ED*, the more nodes and edges are different, indicating the stronger privacy preservation.

2) Privacy analysis

As described in Table1, as ε descends from 2 to 0.2, the *ED* of *LDPGG* algorithmin raises from 5297 to 6794. In other three data sets, there are the same results. The results show that the smaller the privacy budget, the stronger the privacy preservation of the *LDPGG* algorithm. This indicates that the privacy budget can regulate the privacy preservation level of *LDPGG* algorithm. In addition, the *ED* value of *LDPGG* algorithm is larger than the *k*-anonymity-GM algorithm in all data sets.For example, in Enron dataset, when the privacy budget ε is 2, the *ED* value of the *LDPGG* algorithm is 66881, while the *ED* value of the *k*-anonymity-GM algorithm with *k*=10 is 44646. However, in all data sets, compared with the *LDPGen* algorithm, the *ED* value of the *LDPGG* algorithm is smaller. Therefore, the *LDPGG* algorithm has stronger privacy preservation strength than the *k*-anonymity-GM algorithm, but is weaker than the *LDPGen* algorithm.

Fig.3 illustrates the comparison of *ED* values in three algorithms on the Facebook data set. The result shows that the *ED* value of the *k*-anonymity-GM algorithm is the smallest among the three algorithms. As ε increases, the *ED* values of both *LDPGG* and *LDPGen* algorithms decrease, with the *ED* values of the *LDPGG* algorithm being smaller than those of *LDPGen* algorithm. This indicates that the privacy preservation strength of the *LDPGG* algorithm is lower than that of the *LDPGen* algorithm, but stronger than the *k*-anonymity-GM algorithm. Fig.4 shows a comparison of the *ED* values in three algorithms on the Enron data set. The results show that the *ED* value of the *LDPGG* algorithm is larger than that of the *k*-anonymity-GM algorithm and smaller than that of the *LDPGen* algorithm.

TABLE I: The value of *ED* in three algorithms

algorithms	***parameter***	**Blogs1224**	**Facebook 4039**	**Enron36692**	**Facebook 63731**
LDPGG	ε=0.2	6786	32887	153523	278623
LDPGG	ε=0.5	6332	28534	139812	246834
LDPGG	ε=1	5857	24976	113112	215187
LDPGG	ε=1.5	5513	22234	99179	194831
LDPGG	ε=2	5297	20125	76881	189432
k-anonymity-GM	k=3	3301	6932	32876	57742
k-anonymity-GM	k=5	3467	7469	35915	58790
k-anonymity-GM	k=7	3755	9277	39832	60467
k-anonymity-GM	k=10	4147	12987	44723	62423
LDPGen	ε=0.2	8189	43285	212863	472742
LDPGen	ε=0.5	7654	39523	188903	411465
LDPGen	ε=1	7265	35231	155613	379243
LDPGen	ε=1.5	6934	33643	135462	338786
LDPGen	ε=2	6752	31132	114327	318215

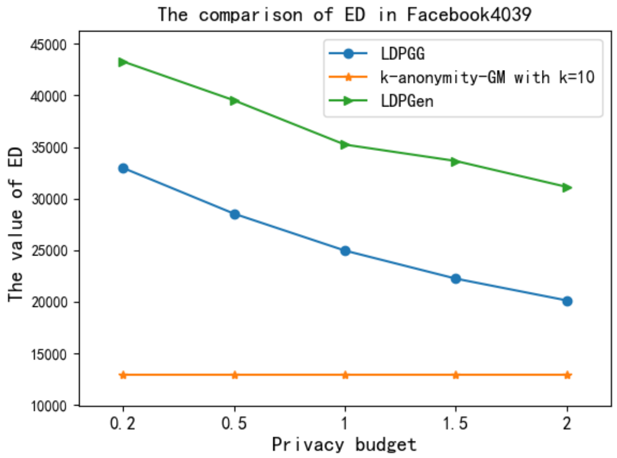

Fig. 3: The comparison of *ED* in three algorithms on FaceBook4039

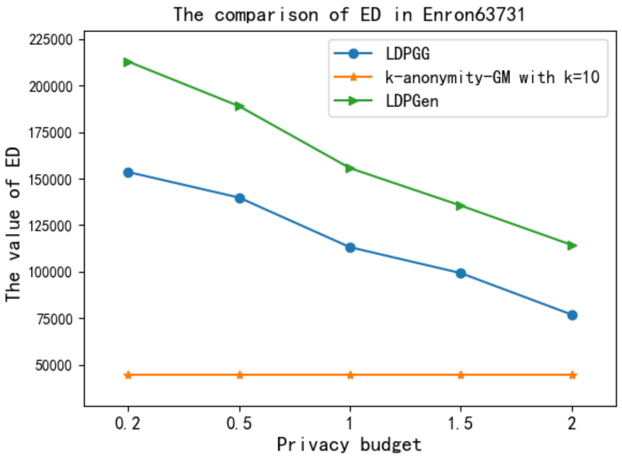

Fig. 4: The comparison of *ED* in three algorithms on Enron36692

This indicates that the *LDPGG* algorithm achieved the same conclusion on the Enron data set as on the Facebook data set.

To summarize, the privacy budget ε determines the strength of privacy preservation for the *LDPGG* algorithm. Given a privacy budget, the *LDPGen* algorithm provides stronger privacy preservation than *LDPGG*. The reason is that the *LDPGen* algorithm focuses on generating synthetic graphs without paying attention to the link relationships of each node. Therefore, the *LDPGen* algorithm adds a large amount of noise to each node, causing significant perturbation to each node. In the *LDPGG* algorithm, the perturbation of each node is limited in 2-hop subgraph of each node. In addition, the properties of unbiased estimate and the similarity between two nodes are utilized to maintain the link relationship of each node. In the end, the *LDPGG* algorithm provides privacy preservation while maintaining data utility.

C. Data utility evaluation

1) Data utility measurement

The *Utility* is usually applied to evaluate the data utility of the algorithm, and it is defined as follows.

$$Utility = (1 - \frac{(|PM - RM|)}{RM}) \times 100\%$$

where *PM* represents one graph metric in a synthetic graph generated by different algorithms, *RV* denotes one real metrics in a original graph.

Note that the greater the *Utility*, the better the data utility of this .

To measure the *Utility* of the algorithm, four graph metrics are used, such as *NE*(the number of edges),*AD*(the average degree of nodes), S_{Diam}(the diameter of the graph) and S_{APD}(the average shortest distance among nodes)

2) *Data utility analysis*

As illustrate in Fig.5, in the Enron data set, the maximum *Utility* of *NE* in *LDPGG* algorithm reaches 76% with ε being 2 while the minimum *Utility* of *NE* in *LDPGG* algorithm is 70% with ε being 0.2. Thus, the average *Utility* of *NE* is about 73%. As shown in Fig.6, in the Facebook4039 data set, the maximum *Utility* of *AD* in *LDPGG* algorithm is 76%, the minimum *Utility* of *AD* reaches 70%. Therefore, the average *Utility* of *AD* is about 73%.

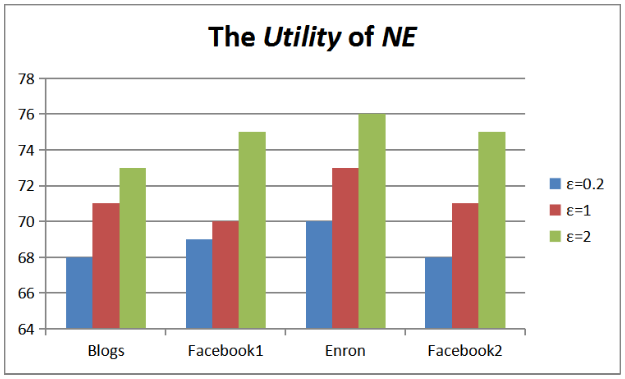

Fig. 5: the *Utility* of *NE*

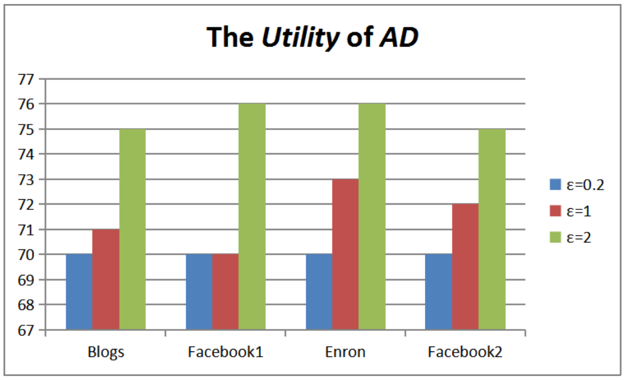

Fig. 6: the *Utility* of *AD*

As illustrate in Fig.7, in the Enron email network data set, as ε increases, the *Utility* of S_{Diam} in *LDPGG* algorithm and *LDPGen* algorithm simultaneously rises,while the *Utility* of S_{Diam} in k-anonymity-GM algorithm maintains unchanged. The result indicates that the *Utility* of S_{Diam} is determined by ε. Compared with the *LDPGen* algorithm, the *Utility* of S_{Diam} in *LDPGG* algorithm is larger. However, the *Utility* of S_{Diam} in *LDPGG* algorithm is smaller than that in k-anonymity-GM algorithm. Therefore, the utility of the *LDPGG* algorithm is better than the *LDPGen* algorithm but is lower than the k-anonymity-GM algorithm.

As shown in Fig.8, in the Facebook63731, as ε is 2, the *Utility* of S_{ASD} in *LDPGG* algorithm almost reaches 70%, which is close to the *Utility* of S_{ASD} in k-anonymity-GM algorithm. In addition, the *Utility* of S_{ASD} in *LDPGG* algorithm is larger than that in *LDPGen* algorithm. According to the result of comparison in three algorithms, although the data utility of the *LDPGG* algorithm is weaker than the k-anonymity-GM algorithm, but it is better than the *LDPGen* algorithm. To sum up, the data utility of the *LDPGG* algorithm is effective.

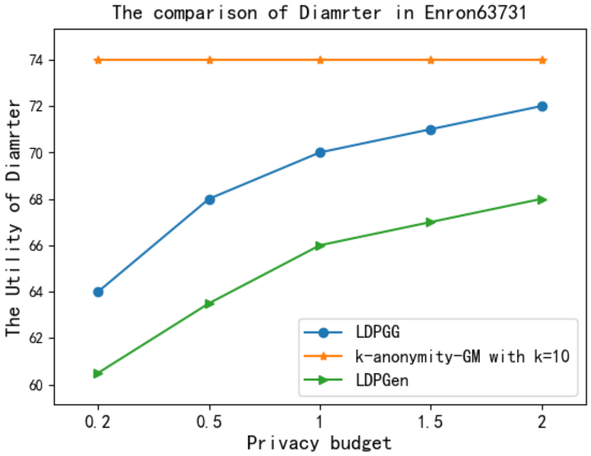

Fig. 7: The comparison of algorithms on S_{Diam}

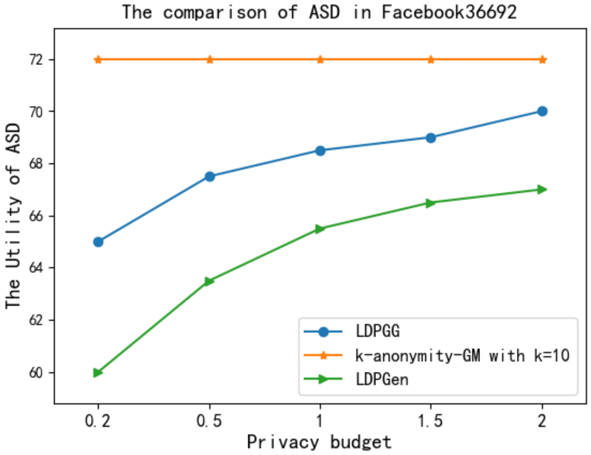

Fig. 8: The comparison of algorithms on S_{ASD}

To sum up, regardless of the privacy budget, the data utility of the *LDPGG* algorithm is weaker than that of graph generation algorithms based on k-anonymity, indicating that local differential privacy provides stronger privacy preservation than graph modification. Compared with the *LDPGen* algorithm, the *LDPGG* algorithm achieves better data utility due to node encoding based on its 2-hop subgraph, the property of unbiased estimate and the similarity between two nodes. Therefore, the *LDPGG* algorithm maintains the data utility of the synthetic graph while preserving the link privacy of nodes.

VII. Conclusion

In the privacy preservation of graph data in social networks, generating synthetic graphs provides a universal protection paradigm. A local differential privacy graph generating method is proposed to address the link privacy in social networks.

The entire method consists of two parts: the client end and the data collection end. On the client side, each node achieves link privacy protection by encoding 2-hop subgraphs and randomly responding to perturb the node structure. The data collection end corrects the perturbations of each node and reconstructs the node subgraph, ultimately merging the subgraphs to generate a synthetic graph. In the implementation of the method, node encoding based on 2-hop sub-graph algorithm, node sequence random response algorithm, and synthetic graph generation algorithm were designed. In addition, theoretical analysis and experimental results indicate that graph generation methods based on local differential privacy not only achieve link privacy protection for each node, but also have effective data utility.

In the future, it is important to achieve the tradeoff between privacy preservation and data utility in local differential privacy. In addition, it is a great work to preserve more complex networks through local differential privacy.

ACKNOWLEDGMENT

This research was funded by the National Natural Science Foundation of China(62372076), the Fundamental Research Funds for the Central Universities (GK202405003), the Shaanxi Provincial Natural Science Foundation project (2024JC-YBMS-543), the research project of Shangluo University (21SKY126) and Shangluo Universities Key Disciplines Project, Discipline name:Mathematic.

REFERENCES

[1] M. Umar, J. Wang, L. Liu, Z. Guo and S. Wang, “Physical layer authentication in the internet of vehicles based on signal propagation attribute prediction, ”*Journal of Networking and Network Applications*, vol.3, no.1, pp.1-10, 2023

[2] A. Rejeb, K. Rejeb, H. Treiblmaier, A. Appolloni, S. Alghamdi, Y. Alhasawi and M. Iranmanesh, “The Internet of Things (IoT) in healthcare: Taking stock and moving forward, ”*Internet of Things*, vol.22, pp.1-29, 2023.

[3] Z. Guo, J. He, Y. Zhang, S. Zhao, Y. Shen and X. Jiang, “Covert Communications in Satellite Internet: A Survey, ”*Journal of Networking and Network Applications*, vol.2, no.3, pp.120-128, 2022.

[4] T. Hennig-Thurau, D. N. Aliman, A. M. Herting, G. P. Cziehso, M. Linder and R. V. K¨¹bler, “Social interactions in the metaverse: Framework, initial evidence, and research roadmap, ”*J ACAD MARKET SCI*, vol.51, no.4, pp.889-913, 2023.

[5] C. Wu, “Data privacy : From transparency to fairness,” *TECHNOL SOC*, vol.76, pp.1-10, 2024.

[6] M. Bhattacharya, S. Roy, S. Chattopadhyay, A. K. Das and S. Shetty, “A comprehensive survey on online social networks security and privacy issues: Threats, machine learning-based solutions, and open challenges. Security and Privacy, ” *SECUR PRIVACY*, vol.6, no.1, pp.1-28, 2023.

[7] G.J.X. Dance, M.LaForgia and N.Confessore, “As Facebook raised a privacy wall, it carved an opening for tech giants, ”*The New York Times*, vol.18, pp.1-9, 2018.

[8] A. Zulkifli, “TikTok in 2022 : revisiting data and privacy,” *Computer*, vol.55, no.6, pp.77-80, 2022.

[9] H.Wang, Z.Cui, R.Liu, L.Fang and Y.Sha, “A multi-type transferable method for missing link prediction in heterogeneous social networks, ”*IEEE T KNOWL DATA EN*, vol.35, no.11, pp.10981-10991, 2023.

[10] A.Narayanan and V.Shmatikov, “De-anonymizing social networks,”in *Proceedings of the 30th IEEE symposium on security and privacy*, 2009, pp.173-187.

[11] S.A.Myers, A.Sharma, P.Gupta and J.Lin, “Information network or social network? The structure of the Twitter follow graph,”in *Proceedings of the 23rd International Conference on World Wide Web*,2014,pp.493-498.

[12] H.Jiang, J.Pei, D.Yu, J.Yu, B.Gong and X.Cheng, “Applications of differential privacy in social network analysis : A survey,” *IEEE T KNOWL DATA EN*, vol.35, no.1, pp.108-127, 2023.

[13] Y.Li, M.Purcell, T.Rakotoarivelo, D.Smith, T.Ranbaduge and K.S.Ng, “Private graph data release: A survey,” *ACM COMPUT SURV*, vol.55, no.11, pp.1-39, 2023.

[14] S.Zhang, W.W.Ni and N.Fu, “Differentially private graph publishing with degree distribution preservation, ”*COMPUT SECUR*, vol.106, pp.1-17, 2021.

[15] J.Abawajy, M.I.H.Ninggal and T.Herawan, “Vertex re-identification attack using neighbourhood-pair properties, ”*CONCURR COMP-PRACT E*, vol.28, no.10, pp.2906-2919, 2016.

[16] Q.Ye, H.Hu, M.H.Au, X.Meng and X.Xiao, “LF-GDPR:A framework for estimating graph metrics with local differential privacy, ”*IEEE T KNOWL DATA EN*, vol.34, no.10, pp.4905-4920, 2022.

[17] P.Liu, Y.X.Xu, Q.Jiang, Y.Tang, Y.Guo, L.E.Wang and X.Li, “Local differential privacy for social network publishing, ” *Neurocomputing*, vol.391, pp.273-279, 2020.

[18] Z.Qin, T.Yu, Y.Yang, I.Khalil, X.Xiao and K.Ren, “Generating synthetic decentralized social graphs with local differential privacy,” in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp.425-438.

[19] C.Wei, S.Ji, C.Liu, W.Chen and T.Wang, “Asgldp:Collecting and generating decentralized attributed graphs with local differential privacy, ”*IEEE T INF FOREN SEC*, vol.15, pp.3239-3254, 2020.

[20] L.Hou, W.Ni, S.Zhang, N.Fu and D.Zhang, “PPDU:dynamic graph publication with local differential privacy,” *KNOWL INF SYST*, vol.65, no.7, pp.2965-2989, 2023.

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

[22] K.Hayawi, P.H.Ho, S.S.Mathew and L.Peng, “*k*-Anonymous Query Scheme on the Internet of Things: a Zero Trust Architecture,” *Journal of Networking and Network Applications*, vol.1, no.3, pp.88-102, 2021.

[23] J.Casas-Roma, J.Herrera-Joancomart¨ª and V.Torra, “*k*-Degree anonymity and edge selection: Improving data utility in large networks,” *KNOWL INF SYST*, vol.50, no.2, pp.447-474, 2017.

[24] R.Mortazavi and S.H.Erfani, “GRAM:An efficient (*k*, *l*) graph anonymization method,” *EXPERT SYST APPL*, vol.153, pp.1-9, 2020.

[25] W.Ren,K. Ghazinour and X.Lian, “*kt*-Safety: Graph Release via *k*-Anonymity and *t*-Closeness,” *IEEE T KNOWL DATA EN*, vol.35, no.9, pp.9102-9113, 2022.

[26] X.Ding, C.Wang, K.K.R.Choo and H.Jin, “A novel privacy preserving framework for large scale graph data publishing,” *IEEE T KNOWL DATA EN*, vol.33, no.2, pp.331-343, 2019.

[27] N.Yazdanjue, M.Fathian and B.Amiri, “Evolutionary algorithms for *k*-anonymity in social networks based on clustering approach,” *The Computer Journal*, vol.63, no.7, pp.1039-1062, 2020.

[28] Y.Tian, Z.Zhang, J.Xiong, L.Chen, J.Ma and C.Peng, “Achieving graph clustering privacy preservation based on structure entropy in social IoT,” *IEEE INTERNET THINGS*, vol.9, no.4, pp.2761-2777, 2021.

[29] P.Boldi, F.Bonchi, A.Gionis and T.Tassa, “Injecting uncertainty in graphs for identity obfuscation,” *Proceedings of the VLDB Endowment*, vol.5, no.11, pp.1376-1387, 2012.

[30] P.Mittal, C.Papamanthou and D.Song, “Preserving Link Privacy in Social Net-work Based Systems,” in *Proceedings of 20th Annual Network and Distributed System Security Symposium*, 2013, pp.1-10.

[31] H.H.Nguyen, A.Imine and M.Rusinowitch, “Anonymizing Social Graphs via Uncertainty Semantics,” in *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*, 2015, pp.495-506.

[32] J.Hu, J.Yan, Z.Q.Wu, H.Liu and Y.H.Zhou, “A Privacy-Preserving Approach in Friendly-Correlations of Graph Based on Edge-Differential Privacy,” *J INF SCI ENG*, vol.35, no.4, pp.821-837, 2019.

[33] C.Dwork, “Differential Privacy,” in *Proceedings of the 33rd International Col-loquiium on Automata, Languages and Programming*, 2006, pp.1-12.

[34] J.Xing and X.Ma, “DP-gSpan: A Pattern Growth-based Differentially Private Frequent Subgraph Mining Algorithm,” in *Proceedings of 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2021, pp.397-404.

[35] X.Ding, S.Sheng, H.Zhou, X.Zhang, Z.Bao, P.Zhou and H.Jin, “Differentially private triangle counting in large graphs,” *IEEE T KNOWL DATA EN*, vol.34, no.11, pp.5278-5292, 2021.

[36] S.Lan, H.Xin, W.Yingjie, G.Yongyi, “Sensitivity reduction of degree histogram publication under node differential privacy via mean filtering,” *CONCURR COMP-PRACT E*, vol.33, no.8, pp.1-8, 2021.

[37] X.Jian, Y.Wang and L.Chen, "Publishing graphs under node differential privacy," *IEEE T KNOWL DATA EN*, vol.35, no.4, pp.4164-4177, 2021.

[38] H.Huang, D.Zhang, F.Xiao, K.Wang, J.Gu and R.Wang, "Privacy-preserving Approach PBCN in Social Network with Differential Privacy," *IEEE TRANS NETW SERV*, vol.17, no.2, pp.931-945, 2020.

[39] V.Karwa and A.B.Slavkovi¡äc, "Differentially private graphical degree sequences and synthetic graphs," in *Proceedings of International Conference on Privacy in Statistical Databases*, 2012, pp.273-285.

[40] T.Gao and F.Li, "Sharing social networks using a novel differentially private graph model," in *Proceedings of 16th IEEE Annual Consumer Communications & Networking Conference*, 2019, pp.1-4.

[41] J.Imola, T.Murakami and K.Chaudhuri, "Locally Differentially Private Analysis of Graph Statistics," in *Proceedings of 30th USENIX Security Symposium*, 2021, pp.983¨C1000.

Jun Yan received the M.S. degree in College of Earth Exploration Science and Technology from Jilin University and the Ph.D. degree in Computer software and theory from Shaanxi Normal University. His research interests include network security and privacy preserving.

Yijun Zhang received her B.E. degree in the Mathematics and Information Science from Shaanxi Normal University, Shaanxi, China. Now she is currently pursuing the M.S. in Shaanxi Normal University. Her research interests include information security and privacy preserving.

Yi Tian received his B.S. degree in 2005 from kunming university of science and technology, China, and received his M.S. in 2011 from northwest university, China. He is an associate professor of Shangluo College, China. His research interests include computer communications networks and wireless networks.

Yihui Zhou received her B.E. degree, M.S. degree and Ph.D. degree in College of Mathematics and Information Science from Shaanxi Normal University, Shaanxi, China, in 2003, in 2006 and in 2009, respectively. Now she is a lecturer in School of Computer Science, Shaanxi Normal University. Her research interests include information security and privacy preserving.

Laifeng Lu received M.S.degree and Ph.D.degree in Computer system architecture from Xi'dian University, Shaanxi, China. Now she is an associate professor in Shaanxi Normal University. Her research interests include security and privacy protection.