JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 35, 821-837 (2019)
DOI: 10.6688/JISE.201907_35(4).0007

A Privacy-Preserving Approach in Friendly-Correlations of Graph Based on Edge-Differential Privacy

JING HU, JUN YAN, ZHEN-QIANG WU+, HAI LIU AND YI-HUI ZHOU
School of Computer Science
Shaanxi Normal University
Xi'an, 710119 P.R. China
E-mail: {hujing; yanrongjunde; zqiangwu; liuhai; zhouyihui}@snnu.edu.cn

It is a challenging problem to preserve friendly-correlations between individuals when publishing social network data. To alleviate this problem, uncertain graph has been presented recently. The main idea of uncertain graph is converting an original graph into an uncertain form, where the friendly-correlations of the graph are associated with probabilities. However, the existing methods of uncertain graph lack rigorous guarantees of privacy and rely on the assumption of adversary's knowledge. In this paper, we introduced a general model for constructing uncertain graphs. Then, we proposed an Uncertain Graph based on Differential Privacy algorithm (UGDP algorithm) under the general model which provides a rigorous privacy guarantee against powerful adversaries, and we define a new metric to measure privacy for different algorithms. Finally, we evaluate some uncertain algorithms in privacy and utility, the result shows that UGDP algorithm satisfies edge-differential privacy and the data utility is acceptable. The conclusions are that the UGDP algorithm has better privacy preserving than the (k, ε)-obfuscation algorithm, and better data utility than the RandWalk algorithm.

Keywords: social network data, data-correlations, privacy preserving, uncertain graph, differential privacy

1. INTRODUCTION

With the fast development of social networks, people can make friends and share their mood state whenever and wherever to enhance their friendship on social software, and their activities on social software accumulated large amounts of personal privacy data. However, when people are enjoying the convenience of social network, people's sensitive information was exposed, such as personal account, password and social correlations. LinkedIn, a well-known users' friendly-correlations network, which suffered a massive data breach accident in 2012. Almost after four years, a hacker under the nickname "Peace" is offering for sale what he/she claims to be the database of 167 million emails and hashed passwords, which included 117 million already cracked passwords, belonging to LinkedIn users [1], the correlations about users' were leaked. Therefore, how to provide the convenience to users while protecting their privacy is a challenging problem of social networks.

The process of privacy preserving in social networks has three steps which can be seen in Fig. 1. The first step is data processing, the data collector stores user behavior data in a dataset then they convert those data into network graph data. There exists different kinds of privacy preserving approaches in second step, researchers use different

Received December 30, 2017; revised September 6, 2018; accepted November 21, 2018.
Communicated by Ralf Klasing.

[+] Corresponding author.

methods to manipulation data, for example, uncertain graphs is a method that injects probability to every edge of graph. The third step is publishing protected data, the released data are perturbation values after privacy preserving approaches, for example, each edge in uncertain graph is a probability value rather than an exact value.

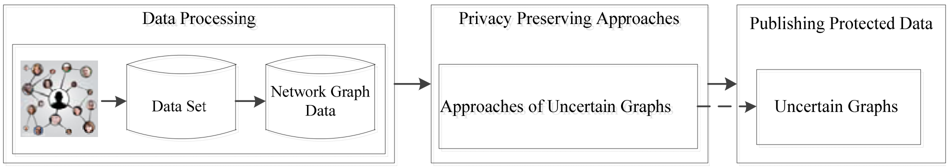
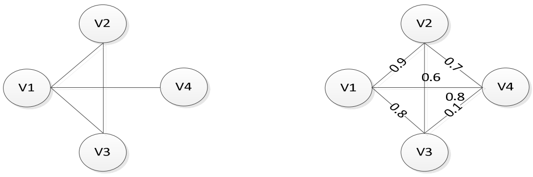

Data Processing
Data Set
Network Graph Data
Privacy Preserving Approaches
Approaches of Uncertain Graphs
Publishing Protected Data
Uncertain Graphs

Fig. 1. The process of privacy preserving.

In order to better study the privacy of social network, researchers abstract the social network as a graph where nodes represent individuals and edges between nodes represent the correlations between individuals. And to preserve privacy in social network, several methods have been proposed, which can be summarized into five main categories: (1) Methods that remove the identities of the nodes before publishing the actual graph or replace the identities of the nodes with synthetic identities; (2) Methods that modify(add, delete or switch edges/nodes) the graph; (3) Methods that generalize vertices and edges into partitions as super-vertices and super-edges; (4) Methods that inject uncertainty into the graph; (5) Methods that provide privacy-aware computation [2] like differential privacy. In this paper, we discuss uncertain graph methods.

The uncertain graph methods have exploited the semantics of graphs to preserve privacy. The main idea is to convert a deterministic graph into an uncertain form. For a more intuitive understanding of the uncertain graph, we give an example, as illustrated in Fig. 2. Fig. 2 (a) is an original graph, Fig. 2 (b) is an uncertain graph obtained by the transformation and modification of the original graph, and each edge of the graph is accompanied by a corresponding probability. We assume that the probabilities of edges are independent. When the uncertain graph (b) is released, due to the uncertainty of the edges, the probability of the attacker getting the original structure is equal to p, where p = 0.9*0.8* 0.8*0.6*(1–0.7)*(1–0.1) = 0.09. Therefore, the released uncertain graph can achieve privacy preserving for the original graph.



(a) (b)

Fig. 2. An example of an uncertain graph.

According to different ways of injecting uncertainty into the original graph, uncertain graph methods can be summarized as (k, ε)-obfuscation algorithm [3], Rand-Walk algorithm [4], Maximum Variance algorithm [5], Maximum Variance algorithm based on uncertain adjacency matrices [6]. However, those methods lack rigorous guarantees of privacy and rely on the assumption of adversary's knowledge adversaries can attack it by malicious background knowledge.

Differential privacy is proposed in [7] to solve the problem of privacy leaked on database by Dwork, which provides a rigorous privacy guarantee against powerful adversaries. It provides a quantifiable way for the privacy level and makes the privacy protection comparable with different privacy budgets. Soon afterwards, researchers applied differential privacy to preserve privacy of graph data.

In this paper, we extend our work [8] and introduce a model to explain the main idea of uncertain graph. Under the model, an algorithm based on edge-differential privacy is proposed, which provides an efficient, rigorous and quantifiable way for the privacy level of uncertain graphs. The algorithm is proved to satisfy edge-differential privacy. Meanwhile, in order to compare privacy directly between different algorithms, we define edge-entropy to measure algorithm's privacy.

1.1 Our Contribution

In this paper, the main contributions are as follows:

(1) We abstract the process of converting an original graph into an uncertain form as a model and give a brief overview of the model.
(2) We present a UGDP algorithm under the model, and we analyze the privacy and utility of the algorithm. The algorithm is proved to satisfy edge-differential privacy and the data utility of UGDP algorithm is acceptable.
(3) We define edge-entropy to compare privacy between different algorithms.

1.2 Paper Outline

The structure of this paper is organized as follows. Section 2 described related work about privacy preserving methods in social networks. The preliminaries were introduced in Section 3. Section 4 discusses the model and algorithm we proposed. Section 5 outlines algorithm analysis in two aspects, privacy analysis and utility analysis. Finally, we summarized the paper in Section 6.

2. RELATED WORK

As we already mentioned above, methods for preserving privacy in social networks can be broadly classified into five categories: simple anonymization, edge and node modification, generalization approaches, uncertain graph, differential privacy.

In the simple anonymization, the method attempts to break the correlation between the real-world identity and sensitive data. In 2007, Backstrom [9] pointed out that the simple anonymization by removing the identities of the nodes before publishing the actual graph or replace the identities of the nodes with synthetic identities does not always guarantee privacy.

In the edge and node modification methods, there are two kinds of implementation ways including random perturbation and constraints perturbation. In the random perturbation way, Hay [10] proposed a random perturbation technique by distorting structural features which is easy to operate, however the central nodes can also be re-identified easily. Casas-Roma in [11] proposed an algorithm for randomization on graphs with considering the edge's relevance. This method achieves a better trade-off between data utility and privacy level. In the constrained perturbation way, the notion of *k*-degree anonymity was proposed in [12] and Liu devised an edge swap algorithm to construct a *k*-degree anonymous network. On the basis work of Liu, several enhanced approaches have been proposed. A greedy algorithm was proposed in [13] which is more effective than the algorithm proposed by Liu on large real graphs.

In the generalization approaches, Hay [14] applied structural generalization approaches using the size of a partition to ensure node anonymity. Stokes and Torra [15] proposed two methods for graph partitioning using the Manhattan distance and the 2-path similarity as measures to create the clusters which group vertices into partitions of *k* or more elements.

In the uncertain graph methods, Boldi in [3] introduced the concept of (k, ε)-obfuscation where k is the desired obfuscation level and ε is a tolerance parameter, and proposed an anonymization approach based on injecting uncertainty. This approach has high impact on node privacy by pursuing minimum standard deviation σ, whereas, a re-identification attack like rounding techniques can easily reveal the true graph. A RandWalk algorithm was introduced by Mittal in [4] to construct the uncertain graph. This method suffers from high lower bounds for utility error despite its excellent privacy-utility trade-off. An approach that not only provides better tradeoff between privacy and utility, but also describes a quantifying framework for graph anonymization based on Maximum Variance was introduced by Nguyen in [5]. The same authors proposed a second approach [6] that adopted a generalized obfuscation model based on uncertain adjacency matrices and kept expected node degrees equal to those in the original graph.

In differential privacy, when researchers applied differential privacy to graph data, two variants of differential privacy were introduced: edge-differential privacy and node-differential privacy [16]. Hay [16] presented an efficient algorithm based on differential privacy for releasing the degree distribution of a network, which provides extremely robust protection, even against powerful adversaries. In node-differential privacy, some algorithms independently about releasing one real-valued statistic in realistic graphs were first presented by Blocki [17], Kasiviswanathan [18] and Chen and Zhou [19]. Day [20] proposed two approaches based on aggregation and cumulative histogram to publish the degree distribution under node-differential privacy. In edge-differential privacy, Karwa [21] presented an efficient algorithm which satisfied edge differential privacy for releasing useful statistical value about graph data while providing rigorous privacy guarantees. Li [22] presented the MB-CI strategy to protect weighted social graphs. The results showed that the MB-CI strategy improved the accuracy and utility of the released data.

3. PRELIMINARIES

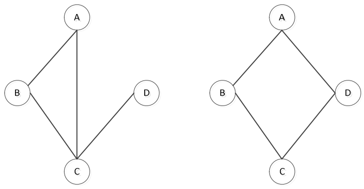
Let $G = (V, E)$ be an undirected graph, where V is the set of vertices and E is the set of edges. V_P denote the set of all C_n^2 unordered pairs of nodes from V, that is, $V_P = \{(v_i, v_j) \mid 1 \le i < j \le n\}$.

Definition 1 (Uncertain graph [3]): Given a graph $G = (V, E)$, an uncertain graph on the vertices of G is a pair $G' = (V, P)$, where P: $V_P \rightarrow [0, 1]$ is a function that assigns probabilities to unordered pairs of vertices.

The uncertain graph G' have the same vertices V of the original graph G. For deterministic graphs, we can assume that the probabilities of all edges are equal to 1.

Definition 2 (Neighboring graph [16]): Given two graphs $G1 = (V1, E1)$ and $G2 = (V2, E2)$, $G1$ and $G2$ are neighbors if $|V1 \oplus V2| + |E1 \oplus E2| = 1$, where $\oplus$ is Exclusive − OR operation.

In this paper, we use Hamming distance to define neighboring graphs. Supposing $V1 = V2$, Definition 2 can be described as that $G1$ and $G2$ are neighbors if $|E1 \oplus E2| = 1$, that is, the Hamming distance between $G1$ and $G2$ is 1.



(a) (b)

Fig. 3. An example of the neighboring graphs.

An example of neighboring graphs is shown in Fig. 3. Fig. 3 (a) is the original graph; according to the definition of the neighboring graphs we can say that Figs. 3 (a) and (b) are neighboring graphs.

Definition 3 (Sensitivity): For any identity mapping f: $G \rightarrow G$ the sensitivity of f is

$$\Delta f = \max_{G1,G2} \|f(G1) - f(G2)\|_1,$$

where $G1$ and $G2$ are neighbors.

In this paper, f is used to query the changes of edges in graph. So, in Fig. 3, the value of Δf is 2.

Due to there is a symbol conflict between the parameters in (k, ε)-obfuscation algorithm and privacy budget ε of differential privacy, we use ε^1 to denote the privacy budget of differential privacy that appears in the following content.

Definition 4 (Differential Privacy): A randomized algorithm A satisfies ε^1-differential privacy if for all $S \subseteq Range(A)$, the following holds:

$$\Pr[A(G1) \in S] \leq \exp(\varepsilon^1) \times \Pr[A(G2) \in S]$$

where $G1$, $G2$ are neighbors and ε^1 is a parameter for privacy level.

The way to satisfy differential privacy is to add noise to the output of a query. The laplace mechanism provides a solution to handle numeric queries and the exponential mechanism can be applied whether a functions output is numeric or not. In our paper, we use laplace mechanism to achieve differential privacy. The definition of laplace mechanism is as follows.

Laplace distribution is shown in Eq. (1),

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

where μ denotes mean parameter, b is a scale parameter and x is a random variable. The cumulative distribution of laplace distribution can be seen in Eq. (2).

$$F(y_i) = \int_{-\infty}^{y_i} g(x)dx \tag{2}$$

Definition 5 (Laplace Mechanism): Given any identity mapping f: $G \rightarrow G$, algorithm A satisfies ε^1-differential privacy if the following holds:

$$A(G) = f(G)+Lap(\Delta f/\varepsilon^1).$$

where Lap($\Delta f/\varepsilon^1$) can be seen in Eq. (2) and $\mu = 0$, $b = \Delta f/\varepsilon^1$. The way by adding laplace noise to achieve differential privacy is known as the laplace mechanism.

Definition 6 (Post-Processing): Let A: $G \rightarrow \mathcal{G}$ be a randomized algorithm that satisfies ε^1-differential privacy. Let K: $\mathcal{G} \rightarrow G'$ be an arbitrary randomized mapping. Then $K \circ A$ is ε^1-differential privacy.

In Definition 6, $\mathcal{G}$ is a noise graph which satisfies ε^1-differential privacy.

4. MODEL AND ALGORITHM

In this section, we first introduce the model of constructing uncertain graphs in detail. Second, we propose an algorithm under the model, meanwhile, we outline our algorithm and give an example to understand our algorithm. Finally, we define edge-entropy to compare privacy between different algorithms.

4.1 A Process Model of Uncertain Graphs

First of all, we abstract a model (see Fig. 4) to explain the main idea of uncertain graph, that is, the process of converting a determinate graph into an uncertain form.

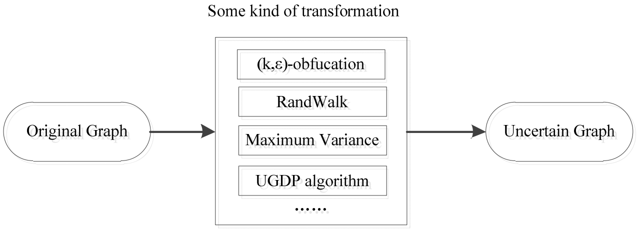


Fig. 4. The model of constructing uncertain graphs.

The model contains three parts. The first part is an original graph. The last part is an uncertain graph. The middle part is some kind of transformations, which are the main part of this model. Different kinds of algorithms can be proposed to construct uncertain graphs, such as (k, ε)-obfuscation algorithm and RandWalk algorithm. In order to support our model, we propose an Uncertain Graph based on Differential Privacy (UGDP) algorithm.

4.2 UGDP Algorithm

In order to better understand the UGDP algorithm we proposed, we refine the model of section 4.1 which can be seen in Fig. 5.

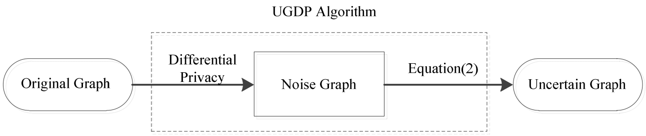


Fig. 5. The UGDP algorithm.

```
Algorithm 1: UGDP algorithm
Input: $G = (V, E)$, sensitivity $\Delta f$ and privacy budget $\varepsilon^1$
Output: $G' = (V, P)$
UGDP($G$, $\varepsilon^1$, $\Delta f$)
    $b \leftarrow \Delta f/\varepsilon^1$
    for $e_i$ in $E$:
        $y_i \leftarrow Lap(b)$
        while $y_i < 0$:
            $y_i \leftarrow Lap(b)$
        $Pr[y_i] \leftarrow F(y_i)$
        $p_i \leftarrow Pr[y_i]$
        adding $p_i$ in $P$
    end
Return $G' = (V, P)$
```

In Fig. 5, we can see that the model of UGDP algorithm has two parts. Firstly, we use laplace mechanism to generate a noise graph from an original graph which satisfies differential privacy. Then we obtain an uncertain graph via cumulative distribution (Eq. (2)) of noise values y_i in noise graph.

We describe the UGDP algorithm in four steps; (1) We achieve differential privacy using laplace mechanism and use $Y = (y_1, y_2, ..., y_i, ..., y_N)$ to denote the noise value which generated through laplace mechanism where N denotes the number of edges in a graph; (2) A noise y_i corresponds to probability p_i where $p_i \in P$, and $p_i = Pr[y_i] = F(y_i)$ (see Eq. (2)); (3) We add probability p_i to the edges of G; (4) We get $G' = (V, P)$ as a output of the algorithm where G' is an uncertain graph.

The UGDP algorithm can be seen in Algorithm 1. The inputs of UGDP algorithm are an original graph, sensitivity Δf and privacy budget ε^1. The scale parameter b in laplace distribution is $\Delta f/\varepsilon^1$ which is outlined in line 1. Lines 3-5 generate the noise value by laplace distribution, which satisfy edge-differential privacy. Lines 6-8 calculate the probability value corresponding to the noise value according to Eq. (2), and gets a set of probability values for each edge of the graph. Finally, UGDP algorithm returns an uncertain graph in line 10, whose edges are assigned with the probability values.

In order to understand our algorithm intuitively, we give an example of the construction process for uncertain graphs. For example, we suppose that Fig. 6 is a real personal relationship network. From Fig. 6, we can know that v_1 has three friends v_2, v_4 and v_5. But in some time, v_1 doesn't want others to know his or her personal relationship, therefore, in order to protect the personal relationship of v_1, we can use UGDP algorithm. The algorithm applied to Fig. 6 including the two steps. First, we add noise to Fig. 6 and we request that this step is to satisfy differential privacy. Second, we use the post processing of differential privacy and transform noise personal relationships into an uncertain form.

So, Fig. 7 is the first step which shows the process of adding noise to the original graph by differential privacy. Fig. 8 is the second step which shows the process of calculates the probability value corresponding to the noise value by using Eq. (2), and converts a noise graph to an uncertain graph.

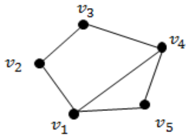


Fig. 6. An example of personal relationship network.

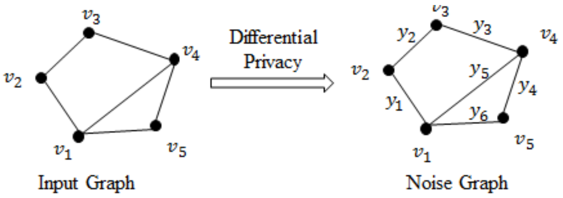


Fig. 7. The process of converting an input graph to a noise graph.

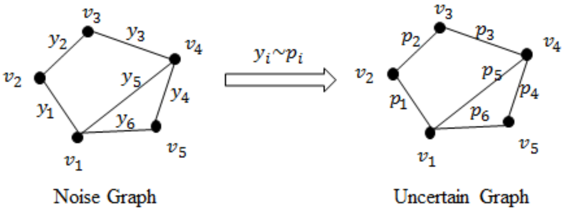


Fig. 8. The process of converting a noise graph to an uncertain graph.

4.3 Edge-entropy

Information entropy is used to measure the amount of information in the information theory. The more ordered a system is, the lower the information entropy is, on the contrary, the more unordered a system is, the higher the entropy of information is. Due to the probabilities of edges in uncertain graphs are stochastic, edge-entropy is introduced to measure the privacy of an algorithm. We use edge-entropy to measure the uncertainty of uncertain graphs, that is to say, the privacy level of uncertain graphs. The definition of edge-entropy is as follows:

$$Ent_e = \sum\nolimits_{e_i \in G'} I_{e_i} \tag{3}$$

where I_{e_i} can be seen in Eq. (4).

$$I_{e_i} = -p(e_i) * \log_2 p(e_i) \tag{4}$$

where $e_i \in G'$ and $p(e_i)$ is the probability of the edge e_i.

The greater Ent_e value, the better privacy level of uncertain graphs, at the same time, it means a better privacy algorithm.

5. ALGORITHM ANALYSIS

In this section, we outline the dataset firstly. Then, we analyze our algorithm in privacy. Finally, we analyze our algorithm in utility.

5.1 Dataset

The experiment data contains two parts, one is the real data sets; the other is the synthetic data sets. The real data sets contain Face-book data with 4039 nodes and 63731 nodes, Enron email network with 36692 nodes and DBLP with 317080 nodes. DBLP is a network of co-authors, and if two authors publish at least one paper together, they will be connected. The synthetic data sets are ER graphs with the number of nodes 200 and 500. The UGDP algorithm, (k, ε)-obfuscation algorithm and RandWalk algorithm are implemented in Python and run on a lenovo computer with the Microsoft Windows 7 operating system, Intel Core i5-4590@ 3.30GHz and 12GB memory.

5.2 Privacy Analysis

There are two parts in this section. Firstly, we prove UGDP algorithm satisfied edge-differential privacy. Then, we use edge-entropy to evaluate the privacy in UGDP algorithm graph , (k, ε)-obfuscation algorithm graph and Rand-walk algorithm graph.

5.2.1 Differential privacy analysis

We give a theorem to illustrate that UGDP algorithm satisfies edge-differential privacy (see Theorem 1).

Theorem 1: UGDP algorithm satisfies ε^1-edge-differential privacy.

***Proof*:** Let $f(\cdot)$ be some identity mapping f: $G \rightarrow G$. $G1$, $G2$ are neighbors and the Hamming distance between $G1$ and $G2$ is 1. Let P_{G1} denote the probability density function of UGDP ($G1, f, \varepsilon^1$), and P_{G2} denote the probability density function of UGDP ($G2, f, \varepsilon^1$). $G3$ is the noise graph obtained during the UGDP algorithm.

We use $p_i \sim y_i$ to denote the correlation of the probability and noise. Due to the post-processing (see Definition 6) technique of differential privacy, we can know that the process of converting a noise graph into an uncertain graph satisfies ε^1-edge-differential privacy. So, in order to prove that UGDP algorithm satisfies ε^1-edge-differential privacy, we only need to prove that the process of converting an original graph into a noise graph satisfies ε^1-edge-differential privacy. The proof process is as follows.

Therefore, UGDP algorithm satisfies ε^1-edge-differential privacy. The smaller privacy budget ε^1, the wider range of the noise value, therefore, we can get better privacy preserving for data.

$$\frac{P_{G1}[G3]}{P_{G2}[G3]} = \frac{P_{G1}[UGDP(G1,f,\varepsilon^1)-f(G1)]}{P_{G2}[UGDP(G2,f,\varepsilon^1)-f(G2)]}$$

$$=\prod_{i=1}^{j}\left(\frac{\exp\left(-\frac{|f(G1)_i-G3_i|}{\Delta f/\varepsilon^1}\right)}{\exp\left(-\frac{|f(G2)_i-G3_i|}{\Delta f/\varepsilon^1}\right)}\right)$$

$$=\prod\nolimits_{i=1}^{j}\exp\left(\frac{|f(G2)_i-G3_i|-|f(G1)_i-G3_i|}{\Delta f/\varepsilon^1}\right)$$

$$\leq\prod\nolimits_{i=1}^{j}\exp\left(\varepsilon^1\cdot\frac{|f(G1)_i-f(G2)_i|}{\Delta f}\right)$$

$$=\exp\left(\varepsilon^1\cdot\frac{\|f(G1)-f(G2)\|_1}{\Delta f}\right)\leq\exp(\varepsilon^1)$$

5.2.2 Privacy comparisons

In order to evaluate the different algorithms in the privacy preserving, we use edge-entropy to measure the algorithm's privacy. The greater edge-entropy value Ent_e, the better privacy algorithm. The experimental data were obtained after averaging many times.

Table 1. The edge-entropy values in different algorithms.

n ε^1 or k or t	$n = 200$	$n = 500$	$n = 4039$	$n = 36692$	$n = 63731$	$n = 317080$
$\varepsilon^1 = 0.01$	1157.59	7209.02	25672.40	53477.33	237742.05	305509.78
$\varepsilon^1 = 0.1$	1137.84	7207.07	25664.06	53510.07	237789.76	305497.12
$\varepsilon^1 = 1$	1184.55	7212.45	25685.42	53499.91	293354.40	376956.78
$k = 10$	35.35	221.36	698.99	1395.07	6252.51	7804.91
$k = 20$	33.42	225.57	697.56	1413.62	6202.99	7871.24
$t = 5$	1959.27	12303.99	42459.23	79936.99	385147.79	398336.13
$t = 10$	1932.32	12285.31	42347.47	79670.20	385047.41	398308.81

For those three uncertain algorithms, due to the combinations of different parameters, there may be exist many possible uncertain graphs. In (k, ε)-obfuscation algorithm, we consider two obfuscation levels where k belongs to {10, 20}, and we give other parameters as follows, tolerance parameter $\varepsilon = 0.1$, multiplier factor $c = 1$, white noise $q = 0.01$. In Rand-Walk algorithm, the parameter t indicates the size of noise. The parameter ε^1 is the privacy budget in UGDP algorithm.

The comparisons can be seen in Table 1. In Table 1, the first to the third row shows the edge-entropy values of UGDP algorithm graph, the fourth line to the fifth row shows the edge-entropy values of (k, ε)-obfuscation algorithm graph and the sixth line to the seventh row shows the edge-entropy values of Rand-walk algorithm. As the number of nodes increase, we can see that the edge-entropy values also increase, which indicates that the uncertainty of the graph is increasing, that is, the more uncertainty are injected in graph. In UGDP algorithm, owing to the randomness of the laplace distribution, the probabilities injected to the edges are also randomness. Therefore, the privacy preserving of the original graph measured between edge-entropy and differential privacy does not necessarily go all the way. What we can see in Table 1 is that the value of edge-entropy in UGDP algorithm graph is bigger than (k, ε)-obfuscation algorithm graph in same datasets, and in addition, the edge-entropy of Rand-Walk algorithm is the largest in three algorithms.

According to the definition of edge-entropy, we can conclude that UGDP algorithm has better privacy preserving than (k, ε)-obfuscation algorithm, but is weaker than Rand-Walk algorithm.

5.3 Utility Analysis

There are two parts in this subsection. Firstly, we define some utility metrics to measure the data utility of the original graphs and uncertain graphs. Then, we use those metrics to evaluate the data utility in UGDP algorithm, (k, ε)-obfuscation algorithm and Rand-walk algorithm.

5.3.1 Utility metrics

Following [3] and [6], we use NE, AD and DV to measure the utility of our algorithm where NE denotes the number of edges in graph, AD denotes the average degree of graph and DV denotes the degree variance of graph. Firstly, let d_1, d_2, ..., d_n denote the degree sequence in a graph G. When G' is an uncertain graph, d_1, d_2, ..., d_n are random variables. Hence, the expected degree of a vertex $v \in V$ is equal to the sum of probabilities of its adjacent edges (see Eq. (5)).

$$d_v = \sum p(i,j) \tag{5}$$

where $i = v$ or $i \neq j$.

Then, we define NE, AD and DV as follows:

- $NE = \frac{1}{2}\sum_{v \in V} d_v$
- $AD = \frac{1}{n}\sum_{v \in V} d_v$

- $DV = \frac{1}{n}\sum_{v\in V}(d_v - AD)^2$

The *NE*, *AD* and *DV* as we defined cannot be directly applied in uncertain graphs. Thus, we use expected statistic to compute them, the definitions are as follows:

- $NE = \frac{1}{2}\sum_{v\in V}\sum_{u\in V\setminus v} p(u,v) = \sum_{e\in V_2} p(e)$
- $AD = \frac{1}{n}\sum_{v\in V}\sum_{u\in V\setminus v} p(u,v) = \frac{2}{n}\sum_{e\in V_2} p(e)$
- $DV = \frac{1}{n}\sum_{v\in V}(d_v - AD)^2$

Next, we use *Utility* to denote data utility achieved by the algorithm. The greater the value is the better the data utility. The definition is as follows:

$$Utility = \frac{PVU}{RVU} \tag{6}$$

where *PVU* denotes the perturbation statistics in uncertain graphs, *RVU* is the real statistics in original graphs.

Finally, we define a new metric I_i to denote the importance of node *i* in the network (see Eq. (7)).

$$I_i = \frac{d_i}{\sum_{i=1}^{n} d_i} \tag{7}$$

where d_i denotes the degree of node *i*.

5.3.2 Utility comparisons

Firstly, we compare UGDP algorithm with (*k*, *ε*)-obfuscation algorithm and Rand-Walk algorithm by *NE*, *AD*, *DV*. The experimental data were obtained after averaging many times.

Table 2 shows the utility metrics in original graph and UGDP algorithm graph. Table 3 shows the utility metrics in original graph and (*k*, *ε*)-obfuscation algorithm graph and Rand-walk algorithm. Comparing to original graph, no matter UGDP algorithm, (*k*, *ε*)-obfuscation algorithm and Rand-Walk algorithm, they all have an impact on the data utility of *NE*, *AD*, *DV*. And the conclusion we made is that (*k*, *ε*)-obfuscation algorithm has better data utility than UGDP algorithm by *NE*, *AD* and *DV*, and UGDP algorithm has better data utility than Rand-Walk algorithm, too.

Next, in order to illustrate the data utility of our algorithm is available we use *Utility* to measure the extent of our data utility achieved. Fig. 9 outlines *NE*'s and *AD*'s Utility.

In Fig. 9 (a), we can see most of data utility *NE* by UGDP algorithm can reach 75.0%, the highest is 75.1%. In Fig. 9 (b), we can see most of data utility *AD* by UGDP algorithm can reach 75.0%, the lowest is 75.0% and the highest is 75.2%. The UGDP algorithm maintains a high utility of degree in uncertain graphs. So, the conclusion we made is that our algorithm's data utility is feasible.

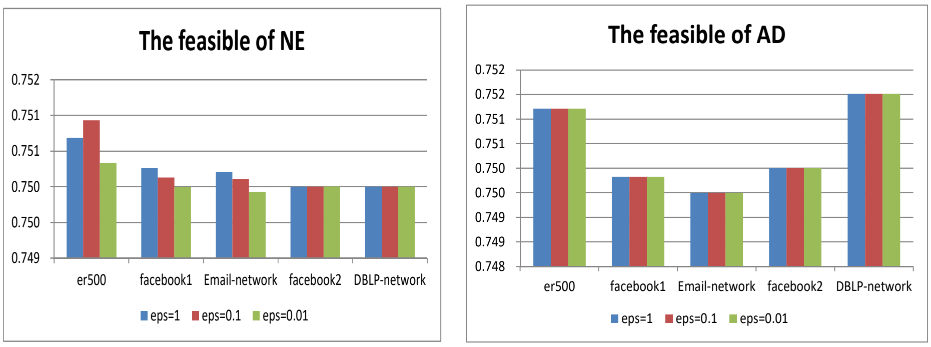
Finally, we use I_i to measure the changes of node importance in obfuscation graphs. Figs. 10 and 11 illustrate the changes of I_i in UGDP algorithm graph and (k, ε)-obfuscation algorithm graph.

Table 2. Utility metrics in UGDP algorithm.

nodes' number	metrics	original graph	ε^1		
			$\varepsilon^1 = 0.01$	$\varepsilon^1 = 0.1$	$\varepsilon^1 = 1$
$n = 200$	*NE*	4049.00	2980.85	2935.48	3057.38
	AD	20.00	29.84	29.35	30.57
	DV	447.89	27.34	23.52	27.88
$n = 500$	*NE*	24818.00	18584.36	18599.10	18593.00
	AD	49.00	74.33	74.39	74.37
	DV	2607.00	62.50	64.36	60.15
$n = 4039$	*NE*	88234	66174.94	66186.89	66198.39
	AD	21.00	32.76	32.77	32.77
	DV	3262.12	1555.22	1556.92	1554.82
$n = 36692$	*NE*	183831.00	137860.15	137893.07	137911.26
	AD	5.00	7.51	7.51	7.52
	DV	1328.41	799.00	798.07	798.86
$n = 63731$	*NE*	817090.00	612833.04	612890.90	612896.75
	AD	12.00	19.23	19.23	19.23
	DV	1785.84	924.60	926.06	925.78
$n = 317080$	*NE*	1049866.00	787429.84	787421.12	787430.32
	AD	3.00	4.96	4.96	4.96
	DV	113.27	59.18	59.19	59.19

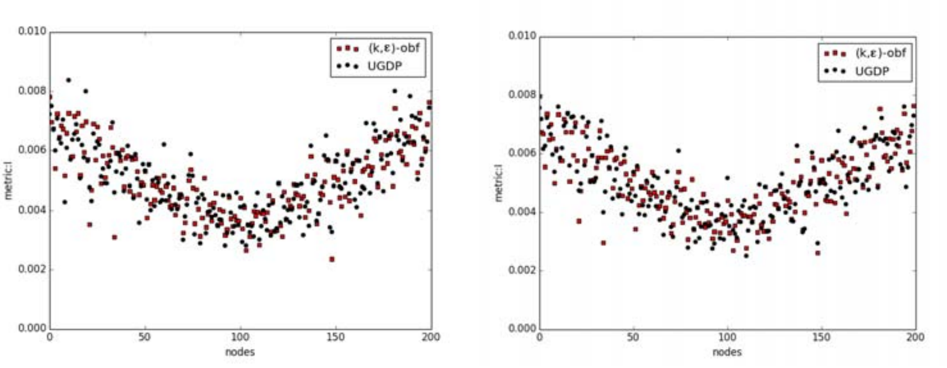
Table 3. Utility metrics in (k, ε)-obfuscation algorithm and Rand-Walk algorithm.

nodes' number	metrics	original graph	k		t	
			$k = 10$	$k = 20$	$t = 5$	$t = 10$
$n = 200$	*NE*	4049.00	2927.57	2985.48	2089.48	2064.08
	AD	20.00	18.13	18.92	10.45	10.32
	DV	447.89	714.35	733.62	136.82	127.83
$n = 500$	*NE*	24818.00	18967.09	18969.48	12625.94	12607.90
	AD	49.00	37.53	37.18	25.25	25.21
	DV	2607.00	4479.33	4511.92	688.08	696.27
$n = 4039$	*NE*	88234	86368.84	86313.12	45176.57	45082.76
	AD	21.00	21.38	21.37	11.23	11.23
	DV	3262.12	3244.85	3245.46	711.13	717.31
$n = 36692$	*NE*	183831.00	183761.76	183730.92	100215.88	99815.99
	AD	5.00	5.00	5.00	2.98	3.09
	DV	1328.41	1328.33	1328.33	387.37	396.34
$n = 63731$	*NE*	817090.00	816286.26	816278.78	425702.09	424627.74
	AD	12.00	12.80	12.80	6.94	6.96
	DV	1785.84	1764.44	1764.44	493.18	498.01
$n = 317080$	*NE*	1049866.00	1049604.00	1049671.30	621827.57	621810.02
	AD	3.00	3.31	3.31	2.03	2.04
	DV	113.27	111.112	111.12	36.09	37.07



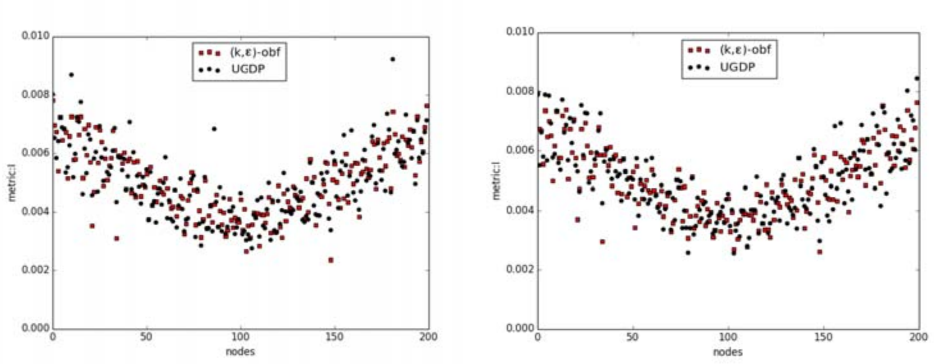
(a) The feasible of *NE*. (b) The feasible of *AD*.

Fig. 9. The data feasible of NE and AD in our algorithm.



(a) $\varepsilon^l = 0.1$ and $k = 10$. (b) $\varepsilon^l = 0.1$ and $k = 20$.

Fig. 10. The comparison of UGDP algorithm graph and (k, ε)-obfuscation algorithm graph using I_i.



(a) $\varepsilon^l = 1$ and $k = 10$. (b) $\varepsilon^l = 1$ and $k = 20$.

Fig. 11. The comparison of UGDP algorithm graph and (k, ε)-obfuscation algorithm graph using I_i.

Due to space limitations and the similarity of plots, we present the results in the circumstances of $k \in \{10, 40\}$ and $\varepsilon^l = \{0.1, 1\}$. In Figs. 10 (a) and (b), we use same privacy budget but different obfuscation levels where $\varepsilon^l = 0.1$ and $k = \{10, 40\}$. We can see that I_i in UGDP algorithm and (k, ε)-obfuscation algorithm is similar. Two algorithms can achieve the similar results in node's importance because I_i of two algorithms distributes in the same probability band. The same conclusions can be made in Figs. 11 (a) and (b) where the privacy budget ε^1 is 1 and obfuscation levels k are {10, 40}. So, given a graph G, we can learn that no matter how much the privacy budget and the obfuscation level are, we have the same utility results by I_i between our UGDP algorithm graph and (k, ε)-obfuscation algorithm graph.

6. CONCLUSIONS

In this paper, we introduced a model for achieving an uncertain graph, and to support our model, we proposed an algorithm to inject uncertainty for edges based on edge differential privacy. The algorithm we proposed not only satisfies the concept of uncertain graphs but also satisfies edge differential privacy. That is, our algorithm satisfies all the characteristics of differential privacy, especially it's strictly provable and rigorous privacy guarantees. And the UGDP algorithm also achieves privacy preserving of uncertain graphs, the relationships between individuals are uncertain and attackers cannot explicitly infer the relationships between them. Meanwhile, we made privacy analysis and utility analysis for three algorithms. The conclusions are that our algorithm has better privacy than (k, ε)-obfuscation algorithm and better utility than Rand-Walk algorithm as well. And we defined some metrics to prove the data utility of our algorithm is feasible. The results we obtained can incite directions for future work.

REFERENCES

1. K. Mohit, "Hacker puts up 167 million LinkedIn passwords for sale," http://thehackernews.com/2016/05/linkedin-account-hack.html, 2016.
2. J. Casas-Roma, J. Herrera-Joancomart, and V. Torra, "A survey of graph-modification techniques for privacy-preserving on networks," *Artificial Intelligence Review*, Vol. 47, 2016, pp. 1-26.
3. P. Boldi, F. Bonchi, A. Gionis, *et al.*, "Injecting uncertainty in graphs for identity obfuscation," in *Proceedings of the VLDB Endowment*, Vol. 5, 2012, pp. 1376-1387.
4. P. Mittal, C. Papamanthou, and D. Song , "Preserving link privacy in social network based systems," in *Proceedings of the 20th Annual Network and Distributed System Security Symposium*, 2013, pp. 1-16.
5. H. H. Nguyen, A. Imine, and M. Rusinowitch, "A maximum variance approach for graph anonymization," *Foundations and Practice of Security*, Vol. 8930, 2014, pp. 49-64.
6. 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.
7. C. Dwork, "Differential privacy," in *Proceedings of the 33rd International Colloquiium on Automata, Languages and Programming*, 2006, pp. 1-12.
8. J. Hu, W. C. Shi, H. Liu, *et al.*, "Preserving friendly-correlations in uncertain graphs using differential privacy," in *Proceedings of International Conference on Networking and Network Applications*, 2017, pp. 24-29.

9. L. Backstrom, C. Dwork, and J. Kleinberg, "Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography," in *Proceedings of the 16th international Conference on World Wide Web*, 2007, pp. 181-190.
10. M. Hay, G. Miklau, D. Jensen, *et al.*, "Anonymizing social networks," in *Proceedings of Very Large Data Bases Conference*, 2007, pp. 173-187.
11. J. Casas-Roma, "Privacy-preserving on graphs using randomization and edge-relevance," in *Proceedings of the 11th International Conference on Modeling Decisions for Artificial Intelligence*, 2014, pp. 204-216.
12. K. Liu and E. Terzi, "Towards identity anonymization on graphs," in *Proceedings of ACM SIGMOD International Conference on Management of Data*, 2008, pp. 93-106.
13. X. Lu, Y. Song, and S. Bressan, "Fast identity anonymization on graphs," in *Proceedings of International Conference on Database and Expert Systems Applications*, Vol. 7446, 2012, pp. 281-295.
14. M. Hay, G. Miklau, D. Jensen, *et al.*, "Resisting structural reidentification anonymized social networks," in *Proceedings of the VLDB Endowment*, Vol. 1, 2008, pp. 102-114.
15. K. Stokes and V. Torra, "On some clustering approaches for graphs," in *Proceedings of IEEE International Conference on Fuzzy Systems*, 2011, pp. 409-415.
16. M. Hay, C. Li, G. Miklau, *et al.*, "Accurate estimation of the degree distribution of private networks," in *Proceedings of the 9th IEEE International Conference on Data Mining*, 2009, pp. 169-178.
17. J. Blocki, A. Blum, A. Datta, and O. Sheffet, "Differentially private data analysis of social networks via restricted sensitivity," in *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*, 2013, pp. 87-96.
18. S. P. Kasiviswanathan, K. Nissim, S. Raskhodnikova, and A. Smith, "Analyzing graphs with node-differential privacy," in *Proceedings of the 10th Theory of Cryptography Conference*, 2013, pp. 457-476.
19. S. Chen and S. Zhou, "Recursive mechanism: towards node differential privacy and unrestricted joins," in *Proceedings of ACM SIGMOD International Conference on Management of Data*, 2013, pp. 653-664.
20. W. Y. Day, N. Li, and M. Lyu, "Publishing graph degree distribution with node differential privacy," in *Proceedings of International Conference on Management of Data*, 2016, pp. 133-138.
21. V. Karwa, S. Raskhodnikova, A. Smith, *et al.*, "Private analysis of graph structure," *ACM Transactions on Database Systems*, Vol. 39, 2014, pp. 1146-1157.
22. X. Y. Li, J. Yang, and Z. L. Sun, "Differential privacy for edge weights in social networks," *Security and Communication Networks*, Vol. 2017, 2017, pp. 1-10.

Jing Hu (胡靜) received the M.S. degree in School of Computer Science, Shaanxi Normal University. Her research interests include network security and privacy preserving.

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

Zhen-Qiang Wu (吳振強) 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.

Hai Liu (劉海) received his B.S. degree (2012) and M.S. degree (2015) from Guizhou University. Currently, he is a Ph.D. student in School of Computer Science, Shaanxi Normal University. His main research interest includes privacy protection.

Yi-Hui Zhou (周異輝) 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.