JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 40, 125-150 (2024)
DOI: 10.6688/JISE.202401_40(1).0008

An Uncertain Graph Method based on Shuffle Model to Preserve Link Privacy of Mobile Social Networks

JUN YAN[1,2], WEN-LI WANG[3], ZHEN-QIANG WU[1,+], LAI-FENG LU[3] AND YI-HUI ZHOU[1]

[1]*School of Computer Science*
[3]*School of Mathematics and Statistics*
Shaanxi Normal University
Xi'an, 710119 P.R. China
[2]*School of Mathematics and Computer Applications*
Shangluo College
Shangluo, 72600 P.R. China
E-mail: {yanrongjunde; zqiangwu[+]; lulaifeng; zhouyihui}@snnu.edu.cn; 1272369927@qq.com

With the rapid development of mobile wireless technology, mobile social networks play a key role in people's online life. However, a large amount of data containing individual relationship information in mobile social networks will result in the leakage of individual privacy. Therefore, how to prevent privacy disclosure of these network data while sharing them to improve services for users is urgent. In order to improve the effectiveness of differential privacy, an uncertain graph method based on the shuffle model is proposed. Especially, the shuffle model is introduced to modify the relationships of nodes, which not only provides differential privacy preserving for the link privacy of nodes, but also improves the data utility of differential privacy. Moreover, node differential privacy is utilized to inject uncertainty on edges, which can reduce perturbations caused by differential privacy. In addition, the exponential mechanism is used to restrict the edge modification in the original graph. The theoretical analysis shows that the uncertain method satisfies differential privacy. The results of experiments show that the uncertain method can effectively preserve link privacy of nodes and maintain data utility.

Keywords: mobile social networks, link privacy, differential privacy, shuffle model, uncertain graph

1. INTRODUCTION

Over past several years, with the fast development of wireless network technologies, such as 4G and 5G [1], mobile social networks (MSNs) have made great progress. First of all, the popularity of MSNs has increased significantly. For example, monthly active users in Facebook, the largest MSN platform in the world, have reached 1.23 billion [2]. Furthermore, all kinds of MSN platforms have transformed from a single interactive platform to a multi-functional service platform, which provides many facilitating services including location service, recommendation service, payment, office, and so on. In the foreseeable future, with the wide application of Internet of Things [3], 6G and other technologies in MSNs, MSNs will be closely related to our life and make it more and more convenient.

However, MSNs also bring a great challenge concerning individual privacy for us [4]. Due to the continuous activities of users on MSNs, MSNs generate a large amount of data about users, which contains a great quantity of sensitive information, such as demo-

Received November 1, 2022; revised December 13, 2022; accepted January 9, 2023.
Communicated by Changqiao Xu.

graphics, diseases, religious/political views, *etc.* If this sensitive information is leaked or illegally used, the privacy of user will be breached, which results in serious security problems. For instance, in 2018, exceeded individual information of 50 million users in Facebook was leaked, which led to the privacy disclosure of 87 million users. As a result, Facebook had to face its damaged reputation as well as a fine of more than $1.6 billion [5]. Thus, when a large amount of data in MSNs is released and used, it is urgent for us to take measures for individual privacy in MSNs.

In general, an MSN usually is described as a social graph, in which a node represents a user while an edge denotes the relationship between two users. Compared with the tabular data, the graph data is more complex, containing a lot of information about the network structure. Especially, if the sensitive relationship of the node is disclosed, the attacker can not only identify the sensitive attributes of the node, but also conduct inference attacks [6]. To tackle the link privacy problem, many graph modification methods have been proposed, including edge and node modification methods, generalization methods [7] and uncertain graph methods [8]. Moreover, to improve data utility, *k*-anonymity methods are designed to achieve graph modification. In [9], an anonymity framework was proposed for the large-scale graph data, in which a *k*-anonymity algorithm based on *k*-decomposition was presented. Unfortunately, these graph modification methods mainly rely on the assumption of adversary's knowledge and are not able to resist attacks based on background knowledge.

As a gold-standard notion of privacy [10], differential privacy can provide strict mathematical guarantee and resist the attacks based on background knowledge. Accordingly, differential privacy has been widely adopted for privacy preserving in graph data [11]. According to application scenarios, there are usually two types of differential privacy: centralized differential privacy (CDP) and local differential privacy (LDP) [12]. In CDP, differential privacy is usually employed to preserve various sensitive values of graphs, including degree distribution [13], graph eigenspectrum [14], and graph centrality measures [15]. Furthermore, to preserve sensitive relationships in graph data, differential privacy is also used to generate a synthetic graph which can better preserve the original graph than these modification methods [16]. However, since CDP requires a trusted third-party information collector, CDP is vulnerable to network attacks or internal attacks [17]. In contrast with CDP, LDP directly adds perturbations on data before data collection, which provides stronger privacy preserving than CDP [18]. As a result, LDP is introduced to preserve social graphs [19, 20]. In particular, the random response mechanism is utilized to generate a synthetic graph which achieve privacy preserving for link privacy [21, 22]. Nevertheless, these synthetic graph generation methods based on LDP fail to preserve important graph properties due to excessive perturbation, thus the adoption and applicability of LDP are limited.

Especially, to overcome the shortcomings of CDP and LDP, a new technology called shuffle model has been developed. Compared with CDP, the shuffle model is able to provide better privacy. Moreover, the shuffle model can achieve better data utility than LDP. Therefore, the shuffle model well realizes the trade-off between privacy protection and data utility and is widely used for tabular data [23] and gradients in federated learning [24]. In [25], the shuffle model is first applied to accomplish privacy preserving of sub-graph counting. Especially, to address the issue caused by high-dimensional data, a wedge shuffling is introduced for triangle and 4-cycle counting tasks. Although, the shuffle effectively preserves the statistic information of graph data, it is a great challenge to generate a synthetic graph through the shuffle model.

Motivated by the work in [25], we focus on utilizing shuffle model to generate a synthetic uncertain graph, which can preserve the sensitive relationship information in original graph while being released for data analysis. In order to achieve this goal, there are some challenges that should be solved. First of all, we must find a way to encode nodes according to their structure and avoid excessive coding dimensions. Then, it is also a difficult task to keep the synthetic uncertain graph similar to the original graph. Thus, we propose an uncertain graph method based on shuffle model. In this method, to achieve privacy preserving, the shuffle model modifies the relationships of nodes, and node differential privacy injects uncertainty on edges of the modified graph. Meanwhile, for maintaining data utility, the privacy amplification of shuffle model is utilized to reduce edge modification and the uncertainty edges can keep the structure of original graph. By far, the exponential mechanism is applied to limit graph modification. Consequently, the uncertain graph method based on shuffle model achieve the link privacy preserving while retaining data utility.

1.1 Our Contribution

In this paper, the main contributions are as follows,

(1) We propose an uncertain graph method based on shuffle model to preserve the graph structure data in mobile social networks. Especially, the shuffle model and the uncertainty are combined to achieve the trade-off between privacy and data utility.
(2) We present an uncertain graph method based on shuffle model (UGSM) algorithm, which can preserve the link privacy in graph data with effective data utility.
(3) We perform experiments using synthetic and real data sets to demonstrate the effectiveness of the proposed algorithm. Compared with other algorithms, the result demonstrates that the proposed algorithm can preserve link privacy of original social graph with a high level of data utility.

1.2 Paper Outline

The organization of this paper is described as follows: In Section 2, we concentrate on the graph modification methods and differential privacy-based methods. Some basic knowledge and definitions are introduced in Section 3. In Section 4, we mainly demonstrate an uncertain graph method based on shuffle model, and describe the algorithms in detail and explicitly analyze the privacy guarantees of the proposed algorithms. Section 5 shows the performance of the proposed method in privacy preserving and data utility. Finally, the conclusion is drawn and the future work is presented in Section 6.

2. RELATED WORK

To preserve graph data, many graph modification methods were firstly presented, including edge and node modification methods, generalization methods, uncertainty graph methods. Then, differential privacy-based methods were also developed, which provided strict privacy guarantee while maintaining data utility.

In edge and node modification methods, in order to maintain the structure of the ori-ginal graph, many *k*-anonymity based methods were presented. In [26], R. Mortazavi developed a (*k*, *l*) graph anonymization method based on edge addition, which achieved a desired data utility. In addition, [27] designed a graph partition-based privacy-preserving scheme, named GPPS, in which the *k*-anonymity was implemented to achieve graph modification by node clustering. In [28], a NaFa algorithm was applied to select all the most appropriate edges and add them to the graph. Therefore, the runtime was effectively reduced and the graph utility was improved simultaneously. Moreover, [9] devised an anonymity framework based on *k*-decomposition, which was specially applied to the protection of the large-scale graph data.

In generalization methods, the structure entropy, which combines data mining with the structural information theory, was adopted by a graph clustering method in [29]. In this method, the correctness and similarity degree of clustering results was an analyzed by normalized structural information and network node partition similarity. To improve privacy preserving, [30] utilized node-LDP to present a privacy-preserving graph clustering method, obtaining more robust privacy preserving while maintaining a higher clustering quality. Compared with two methods mentioned above, uncertainty graph methods were able to provide better data utility for graph modification [31]. Moreover, J. Hu in [32] developed an uncertainty graph method based on edge-differential privacy which enhanced privacy preserving while meeting the requirements of data utility. By far, the random response mechanism was used to get an uncertain graph method, which had more robust privacy preserving [33].

In comparison with the graph modification methods, it was noted that the differential privacy had some advantages that could stop an attack based on background knowledge and provide the rigorous mathematical proof. In CDP, many methods based on differential privacy were widely used to preserve specific sensitive information of graphs and publish private graphs. To publish edge triangle counting under differential privacy, [34] proposed an edge-removal projection algorithm based on edge triangle count sorting and given two methods based on this projection algorithm. In [35], a privacy-preserving mechanism under personalized differential privacy was developed to publish network statistics, such as degree distribution. Due to the application of personalized differential privacy, the data utility of the proposed approach was greatly enhanced. In order to publish a differential private graph, [36] utilized differential privacy to generate a synthetic graph which could preserve the original graph while approximating all cuts of it. Moreover, [37] devised a graph publishing algorithm based on node differential privacy, which maintained the utility on the community structure while providing sufficient privacy preserving.

Particularly, to overcome the weakness of central differential privacy, local differential privacy was applied to preserve graph data [38]. For graph statistics, such as *k*-stars, triangle, 4-cycles, *etc.*, [39] designed a one-round algorithm that was order optimal to count *k*-stars. In addition, a one-round algorithm based on random response was used to preserve triangles. In [40], the first LDP enabled graph metric estimation framework was presented for graph analysis, in which the privacy budget between the two atomic metrics was optimally allocated during data collection. To publish a synthetic Graph, [41] developed a hierarchical random graph model based on local differential privacy, which used the Monte Carlo Markov chain to enhance efficiency and accuracy.

To solve the problem about privacy and utility in CDP and LDP, a new shuffled model was developed in [42, 43]. In [44], a distributed differential private algorithm based on shuffled model was proposed to achieve privacy preserving. Furthermore, the network shuffle based on random walk was proposed in [45], which greatly enhanced the privacy preserving of shuffle model. Especially, the shuffle model was firstly applied to preserve graph data in [25], and it achieved a good trade-off between privacy preserving and data utility in sub-graph counting. In this paper, we utilized the shuffle model to generate a synthetic uncertain graph, which could effectively preserve the link privacy in original graph while retaining data utility.

3. PRELIMINARIES

In this paper, a mobile social network is abstract as a simple undirected graph $G = (V, E)$, where V denotes nodes in G and E represents edges between nodes.

Definition 1 (Differential Privacy [10]): Let $\varepsilon \geq 0$, a randomized algorithm Z is ε-differential privacy if for any two neighboring data sets D and D' and all $S \subseteq Range(Z)$, the following holds,

$$P_r[Z(D)\in S] < e^{\varepsilon} \times P_r[Z(D)\in S], \tag{1}$$

where there is one different record between D and D', ε is a privacy budget. In order to achieve ε-differential privacy, there are two ways: Laplace mechanism and Exponential mechanism.

Definition 2 (Local Differential Privacy [38]): For n users, each user has a record. Let $\varepsilon \geq 0$, a randomized algorithm Z is ε-differential privacy if the probability of Z obtaining the same output result $t*(t*\subseteq Ran(Z))$ on any two records t and t' $(t, t' \in Dom(A))$ satisfies

$$P_r[Z(t)\in t^*] < e^{\varepsilon} \times P_r[Z(t')\in t^*], \tag{2}$$

where $Ran(Z)$ and $Dom(Z)$ are the input and output domains of algorithm Z.

Definition 3 (Privacy amplification by shuffling [42]): Let $n\in N$, $\varepsilon_L \in R_{\geq 0}$, X is a set of input data for each user, $x_i \in X$ is the input data of the ith user, and. $X_{1:n} = (x_1, x_2, \ldots, x_n)\in X^n$. Let $R : X \to Y$ be a local randomizer providing ε-LDP. Let Ms: $X^n \to Y^n$ be an algorithm that given a dataset $x_{1:n}$, computes $y_i = R(x_i)$ for $i\in[n]$, samples a uniform random permutation π over $[n]$, and outputs $y_{\pi(1)}, y_{\pi(2)}, \ldots, y_{\pi(n)}$. Then for any $\delta\in[0,1]$, such that $\varepsilon_L \leq \log(n/16\log(2/\delta))$, Ms provides (ε, δ)-DP, where $\varepsilon = f(n, \varepsilon_L, \delta)$ and

$$f(n, \varepsilon_L, \delta) = \log(1 + \frac{e^{\varepsilon_L} - 1}{e^{\varepsilon_L} + 1}(\frac{8\sqrt{e^{\varepsilon_L}\log(4/\delta)}}{\sqrt{n}} + \frac{8e^{\varepsilon_L}}{n})). \tag{3}$$

Duo to the shuffling, the shuffled data $y_{\pi(1)}, y_{\pi(2)}, \ldots, y_{\pi(n)}$ sent to the data collector provides (ε, δ)-DP, where $\varepsilon \ll \varepsilon_L$.

Definition 4 (Uncertain graph): Given a graph $G = (V, E)$, a function $P: EP \rightarrow [0, 1]$, which assigns probabilities to edges, an uncertain graph $G' = (V, E', EP)$ is obtained by using P, where E' is attained by modifying the E, and EP represents the probabilities of edges. Compared with graph G, the uncertain graph G' has the same nodes as G and has different edges from G. In a deterministic graph, the probabilities of all edges are 1.

Definition 5 (**Neighboring graphs):** Given two graphs $G_a = (V_a, E_a)$ and $G_b = (V_b, E_b)$, if there is one different node between G_a and G_b, $|V_a| = |V_b| + 1$, $E_b \subset E_a$, G_a and G_b are neighboring graphs.

In addition, if there is one different edge between G_a and G_b, $|E_a| = |E_b| + 1$, G_a and G_b are also neighboring graphs.

Definition 6 (Sensitivity): Given two graphs G_a and G_b which are neighborhoods, F is a sequence of queries: $G \rightarrow E$, the sensitivity of F is,

$$\Delta f = \max_{G_a, G_b} \| F(G_a) - F(G_b) \|_1 . \tag{3}$$

The Hamming distance is used to calculate the sensitivity of F. If G_a is different from G_b by one node, the sensitivity of F is d_{max}, where the d_{max} is the maximum degree of nodes in the graph G.

Definition 7 (Laplace Mechanism): Given a sequence of queries F: $G \rightarrow E$, algorithm Z satisfies ε-differential privacy if the following holds,

$$Z(G) = F(G) + lap(\Delta f/\varepsilon) \tag{5}$$

where $lap(\Delta f/\varepsilon)$ represents the Laplace noise with $\mu = 0$, $b = \Delta f/\varepsilon$, the way that makes an algorithm Z satisfy ε-differential privacy by adding Laplace noise is the Laplace mechanism. In the Laplace mechanism, the Laplace noise distribution is shown as follows,

$$n(x) = 1/2b * \exp(-|x - \mu|/b) \tag{6}$$

where μ is a position parameter, b denotes a scale parameter and x is a random variable.

Definition 8 (Exponential Mechanism): Given a data set D, an output range T, a privacy budget ε, and a utility function U: $(D, t) \rightarrow R$, a mechanism M that selects an output $t \in T$ with probability proportional to $\exp(\frac{\varepsilon U(D,t)}{2\Delta U})$ satisfies ε-differential privacy.

Definition 9 (Randomized Response): The randomized response mechanism is defined as follows,

$$P(y_i = k \mid x_i = j) = P_{ij} \tag{7}$$

where x_i is an input which equals j, the probability to output that y_i equals k is P_{ij}. When the value ranges of j and k belong to $\{0,1\}$, $i \subset [1, n]$, n is the number of the inputs. The design matrix P_m of the randomized response is defined as follows,

$$P_m = \begin{pmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{pmatrix}. \tag{8}$$

In the design matrix, the sum of probabilities of each row is 1. Therefore, the design matrix P_m is simplified to

$$P_m = \begin{pmatrix} p_{00} & 1-p_{00} \\ 1-p_{11} & p_{11} \end{pmatrix}. \tag{9}$$

Definition 10 (Randomized Response Satisfying ε-Differential Privacy): Given a para- meter ε, if max $\{P_{00}/P_{10}, P_{00}/P_{01}, P_{01}/P_{11}, P_{10}/P_{11}\} < e^{\varepsilon}$, the randomized response scheme following the design matrix P_m satisfies ε-differential privacy.

Definition 11 (Post-Processing): Assuming a randomized algorithm A that satisfies ε-differential privacy, given a data set D, thus D is preserved by the algorithm A and D' is gained, which is the output of the algorithm A. Let N be an arbitrary randomized mapping, when N is applied on D' to get D'', the algorithm $A \circ N$: $D \rightarrow D''$ satisfies ε-differential privacy.

Definition 12 (Parallel Composition Properties): Given a sequence of algorithms $\{A_1, A_2, ..., A_n\}$, and each algorithm A_i satisfies ε_i-differential privacy, if these algorithms are applied respectively on n disjoint subsets of the input database D, this process is called the parallel composition properties of differential privacy, which satisfies Max ε_i differential privacy.

Definition 13 (Sequential Composition Properties): Given n privacy algorithms A_1, $A_2, \ldots, A_n$, if each A_i ($1 < i < n$) satisfies ε_i-differential privacy, a sequence of $A_1, A_2, \ldots, A_n$ over the same database D satisfies $\sum_{i=1}^{n} \varepsilon_i$ differential privacy.

4. MODEL AND ALGORITHM

In this section, we first introduce the method based on shuffle model to generate an uncertain graph. Then, we propose an algorithm to fulfill the method. Finally, we present the theoretical analysis of this algorithm in detail.

4.1 The Method based on Shuffle Model

In this method, the key task is to utilize shuffle model to generate an uncertain graph which preserves the link privacy of original graphs while providing effective data utility. In particular, the shuffle model realizes the balance between privacy and utility in edge modification. To accomplish privacy preserving, the randomized response mechanism based local differential privacy is introduced for edge modification, which provides stronger privacy preserving than the central differential privacy. On the other hand, the shuffle model uses privacy amplification to improve data utility.

Furthermore, to improve data utility of uncertain graph, the exponential mechanism is applied to select some nodes and only these nodes are modified by shuffle model. Moreover, the uncertainty is injected into the modified graph to keep the structure of original graph. Therefore, the proposed method can effectively realize privacy preserving for original social graphs while maintaining data utility.

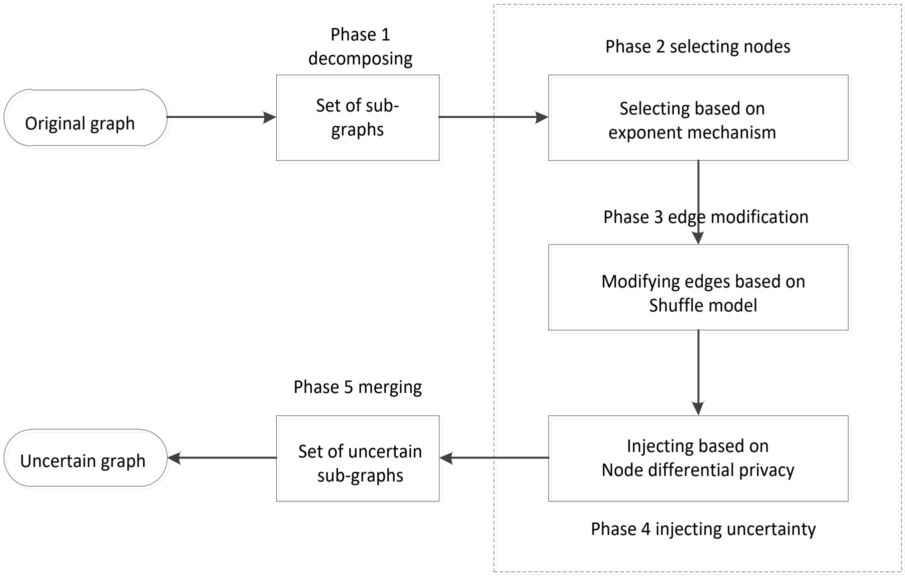

Phase 1
decomposing
Phase 2 selecting nodes
Original graph
Set of sub-graphs
Selecting based on exponent mechanism
Phase 3 edge modification
Modifying edges based on Shuffle model
Phase 5 merging
Uncertain graph
Set of uncertain sub-graphs
Injecting based on Node differential privacy
Phase 4 injecting uncertainty

Fig. 1. The model of proposed method.

The model of proposed method describes in Fig. 1, including five phases. First of all, the original graph is decomposed into many sub-graphs in Phase 1. After all sub-graphs are converted into uncertain sub-graphs, Phase 5 merges them into an uncertain graph. The important part of this method consists of three phases. At the first phase, the key nodes in a sub-graph are selected through exponential mechanism. Then, each key node is encoded according to its 2-hop sub-graph and input into shuffle model, its link relationships are modified by shuffle model. At the last phase, node differential privacy injects uncertainty on edges of the modified sub-graph to get an uncertain sub-graph.

4.2 UGSM Algorithm

To generate an uncertain graph, the UGSM (uncertain graph based on shuffle model) algorithm is presented as follows. In Line 1, a graph G is decomposed into a set of sub-graphs S_s. Then, each sub-graph S_{Gi} is modified by three algorithms from Lines 4 to 7. SNEM (selecting node based on exponential mechanism) algorithm selects important nodes N_{Gi} from S_{Gi} in Line 4. Line 5 modifies the edges of nodes N_{Gi} by using GMSM (graph modification based on shuffle model) algorithm, and UNDP (uncertain graph based on node differential privacy) algorithm generates an uncertain sub-graph S_{mGi} in Line 6. After that, all uncertain subgraphs in S_{Gu} are merged to get an uncertain graph Gu in Line 8. Finally, a differential private uncertain graph is generated.

Algorithm 1: UGSM algorithm

Input: $G = (V, E)$, privacy budget ε_1, ε_2, ε_3
Output: an uncertain graph Gu

```
a set of sub-graphs $S_s$ ← decomposing graph $G$
a set of $S_{Gu}$ = {}
for $S_{Gi}$ in $S_s$:
    a set of nodes $N_{Gi}$ ← SNEM algorithm ($S_{Gi}$, $\varepsilon_1$)
    $S_{nGi}$ ← GMSM algorithm ($S_{Gi}$, $N_{Gi}$, $\varepsilon_2$)
    $S_{mGi}$ ← UNDP algorithm ($S_{nGi}$, $\varepsilon_3$)
    $S_{Gu}$ adding $S_{mGi}$
$Gu$ ← merging $S_{Gu}$
Return an uncertain graph $Gu$
```

4.2.1 SNEM algorithm

In each subgraph S_{Gi}, there are many nodes with different degrees. When noise is added on these nodes, the perturbation of each node is different. For the nodes with small degrees, they suffer more perturbation than those with a big degree. Therefore, when noise is added on the degree value sequence, to reduce the perturbation caused by noise, these nodes with small degrees are deleted and noise is added on nodes with a big degree. In this way, there are two types of disturbances: One is disturbances caused by the deletion, the other is Laplace noise added on the degree value. Given a privacy budget, the exponent mechanism is utilized to get a parameter *m*, which is used to truncate the degree value sequence, so that the minimum noise is added on the degree value sequence.

```
**Algorithm 2:** The **SNEM** algorithm
**Input:** a subgraph $S_{Gi}$, the privacy budget $\varepsilon_1$
**Output:** a set of nodes $N_{Gi}$
$d_s$←$S_{Gi}$    $d_v$←$d_s$
$n$←$|d_v|$
for $m$ in $n$:
    scoring function $-U(G,m)=\sqrt{\sum_{i=m+1}^{n}|d_{v_i}|^2}+\sqrt{2*m}*\frac{\Delta f}{\varepsilon_1}$
selecting $m$ with probability $P_r(m)\propto \exp(-\frac{\varepsilon_1 U(G,m)}{2*\Delta U})$
$d_{vt}$←truncating $d_v$ with $m$
a set of nodes $N_{Gi}$←selecting nodes according to $d_{vt}$
Return $N_{Gi}$
```

In S_{Gi}, the degree sequence is d_s and the degree value sequence sorted from largest to smallest is d_v:[d_{max}, ..., d_{min}]. Then, if d_v is truncated and Laplace noise is added on the truncated d_v, there is a perturbation $Error(d_v)$, which is illustrated as follows,

$$Error(d_v) = DE(d_v) + LE(d_v)$$

where $DE(d_v)$ represents the perturbation caused by the deleted units, $LE(d_v)$ is the Laplace noise added

$$DE(d_v) = E(\sqrt{\sum_{i=m+1}^{n} |d_{vi}|^2})$$

$$LE(d_v) = E(\sqrt{\sum_{i=1}^{m} lap(\Delta f / \varepsilon)^2})$$

$$DE(d_v) + LE(d_v) = \sqrt{\sum_{i=m+1}^{n} |d_{vi}|^2} + \sqrt{2*m}*\frac{\Delta f}{\varepsilon}$$

Here, a query function is f: $f(G) \to d_v$

$\Delta f = |f(G) - f(G')| = |d_v - d'_v| = d_{\max}$

where Δf is the sensitive of a query function f, $d_{\max}$ is the maximum degree of nodes in G and there is only one node difference between G and G'.

Thus, a scoring function U is set up:

$$-U(G,m) = \sqrt{\sum_{i=m+1}^{n} |d_{vi}|^2} + \sqrt{2*m}*\frac{\Delta f}{\varepsilon}$$

In this algorithm, the node differential privacy is applied to achieve differential privacy. Therefore, the ΔU is:

$\Delta U = U(G, m) - U(G', m) = \Delta RE + \Delta LE$

the ΔU is:

$$\Delta RE \le \max\left|\sqrt{\sum_{i=m+1}^{n} |d_{vi}|^2} - \sqrt{\sum_{i=m+1}^{n} |d_{vi}^{'}|^2}\right|$$

$$\le \quad \max\left|\sum_{i=m+1}^{n} |d_{vi}| - \sum_{i=m+1}^{n} |d_{vi}^{'}|\right| \le d_{\max}$$

$$\Delta LE = \Delta f$$

$$\Delta U = \Delta RE + \Delta LE \le 2d_{\max}$$

The probability that the parameter m can be selected is

$$p_r(m) = \frac{\exp(-\frac{\varepsilon_1 * U(G,m)}{2\Delta U})}{\sum_{i=1}^{n} \exp(-\frac{\varepsilon_1 * U(G,i)}{2\Delta U})}$$

Then the parameter m is used to truncate d_v and d_{vt} is obtained. According to d_{vt}, a set of nodes N_{Gi} is gained, which can be utilized to realize the minimal noise perturbation in the original graph.

In Algorithm 2, Line 1 gets the degree value sequence d_v which is sorted from largest to smallest. Then, the number of d_v is obtained in Line 2. From Lines 3 to 5, the exponent mechanism is used to gain a parameter m. According to the m, Line 6 truncates d_v and Line 7 selects nodes according to d_{vt}. In the end, a set of nodes N_{Gi} is obtained.

4.2.2 GMSM algorithm

As shown in Fig. 2, a shuffle model for graph modification is proposed. In this model, each node firstly gives its subgraph information to a randomizer. After disturbing this information through randomized response mechanism, the randomizer sends the obfuscated data to a shuffler in this model. Then, the shuffler randomly permutes the received obfuscated data and releases the shuffled data to the data collector. Finally, this shuffle model provides (ε, δ)-differential privacy for all nodes. In addition, the 2-hop subgraph of each node is encoded as a binary sequence, where 1 represents this node connects one node in this subgraph and 0 denotes there is no edge between this node and another node. When the binary values in the sequence are modified in the model, it means that the edges of the subgraph are added or deleted. Therefore, the shuffle model can achieve graph modification for the input graph.

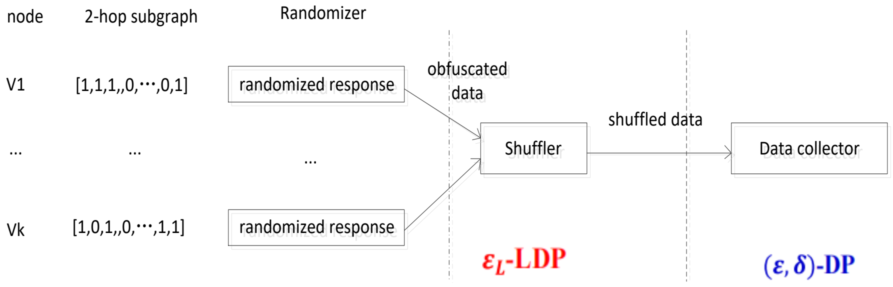


Fig. 2. The shuffle model for graph modification.

Algorithm 3: GMSM algorithm

```
Input: $S_{Gi}$, $N_{Gi}$, $\varepsilon_2$
Output: a modified subgraph $S_{nGi}$
$\varepsilon_L \leftarrow \varepsilon_2$
for $v_i$ in $N_{Gi}$
    2-hop subgraph $S_{Gvi} \leftarrow$ generating from $S_{Gi}$ and $v_i$
    a sequence $S_{vi} \leftarrow$ encoding $S_{Gvi}$
    $SR_{vi} \leftarrow$ randomized response on $S_{vi}$
    a set $S \leftarrow SR_{vi}$
a set $P$ in Data collector $\leftarrow$ a random permutation over $S$
for $v_i$ in $N_{Gi}$
    modified 2-hop subgraph $Sm_{Gvi} \leftarrow p_{vi}$ in $P$
    modifying $S_{Gi}$ according to $Sm_{Gvi}$
$S_{nGi} \leftarrow S_{Gi}$
Return a modified subgraph $S_{nGi}$
```

Especially, although each node uses a large privacy budget ε_L to achieve local differential privacy for itself, owing to the privacy amplification in shuffler, this model provides differential privacy with a small privacy budget ε for all nodes. Since ε_L is far larger than ε, each node can improve data utility of randomized response when ε is given.

Moreover, the shuffle model only randomly permutes the obfuscated data from randomized response. Thus, the shuffle model can provide effective data utility as the data utility of randomized response is improved.

In Algorithm 3, given a privacy budget ε_2, the privacy budget of randomized response ε_L is obtained in Line 1. From Lines 2 to 6, the information of each node is preserved by randomized response mechanism based local differential privacy. After a 2-hop subgraph S_{Gv} is got from S_{Gi} and v_i, it is encoded into a sequence S_{vi} in Lines 3 and 4. Line 5 applies randomized response on S_{vi}, the result is put into a set S in line 6. Then, the shuffler randomly permutes obfuscated data in a set S and sends the result to a set P in Line 7. From Lines 8 to 10, by using p_{vi} in P, modified 2-hop subgraph Sm_{Gvi} is got and it is used to modify S_{Gi}. In the end, a modified subgraph S_{nGi} is obtained in Line 12.

4.2.3 UNDP algorithm

As shown in Fig. 3, the UNDP algorithm consists of three steps. First of all, the Laplace noise is added on each edge of a graph S_{nGi} according to the node differential privacy. In particularity, the node differential privacy is applied to provide better privacy preserving than edge differential privacy. After that, a graph S_{nGi} is transformed into a noised subgraph by adding noise on each edge. Finally, the noise value on each edge is calculated based on the modulo operation. After the calculated result is assigned on this edge, an uncertain graph S_{mGi} is generated. In this algorithm, the modulo operation is to modulo 1 then taking the remainder, so the result of this operation is in [0,1], which is regarded as a probability value.

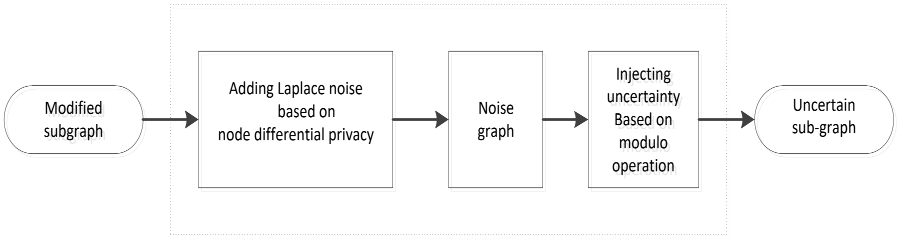

Modified subgraph
Adding Laplace noise based on node differential privacy
Noise graph
Injecting uncertainty Based on modulo operation
Uncertain sub-graph

Fig. 3. The UNDP algorithm.

Algorithm 4: UNDP algorithm

Input: Sn_{Gi}, ε_3

Output: an uncertain subgraph Sm_{Gi}

```
the maximum degree $d_{max}$ in $Sn_{Gi} \leftarrow \Delta f$
a Laplace noise sequence $E_n \leftarrow Lap(\Delta f/\varepsilon_3)$
for $e_i$ in $Sn_{Gi}$
    $e_i \leftarrow E_{ni}$
    $p_i \leftarrow$ the modulo operation ($E_{ni}$)
    if $p_i < 0.5$
        $p_i = 1 - p_i$
    $e_i \leftarrow p_i$
Return an uncertain subgraph $Sm_{Gi}$
```

4.3 The Analysis of UGSM Algorithm

Theorem 1: The SNEM algorithm satisfies ε-differential privacy.

Proof: Given an original graph G, the probability of m to being selected is

$$p_r(m)=\frac{\exp(-\frac{\varepsilon U(G,m)}{2\Delta U})}{\sum_{m'\in o}\exp(-\frac{\varepsilon U(G,m')}{2\Delta U})}.$$

If G and G' are neighborhood graphs, where there is one node difference between them, for any variable m, according to the exponential mechanism, the results are shown as follows,

$$\begin{aligned}\frac{p_r(E(G,m))}{p_r(E(G',m))}&=\frac{\dfrac{\exp(-\frac{\varepsilon U(G,m)}{2\Delta U})}{\sum_{m'\in o}\exp(-\frac{\varepsilon U(G,m')}{2\Delta U})}}{\dfrac{\exp(-\frac{\varepsilon U(G',m)}{2\Delta U})}{\sum_{m'\in o}\exp(-\frac{\varepsilon U(G',m')}{2\Delta U})}}\\
&=\left(\frac{\exp(-\frac{\varepsilon U(G,m)}{2\Delta U})}{\exp(-\frac{\varepsilon U(G',m)}{2\Delta U})}\right)\times\left(\frac{\sum_{m'\in o}\exp(-\frac{\varepsilon U(G',m')}{2\Delta U})}{\sum_{m'\in o}\exp(-\frac{\varepsilon U(G,m')}{2\Delta U})}\right)\\
&\le\exp(\frac{\varepsilon}{2})\times\left(\frac{\sum_{m'\in o}\exp(\frac{\varepsilon}{2})\times\exp(-\frac{\varepsilon U(G,m')}{2\Delta U})}{\sum_{m'\in o}\exp(-\frac{\varepsilon U(G,m')}{2\Delta U})}\right)\\
&\le\exp(\frac{\varepsilon}{2})\times\exp(\frac{\varepsilon}{2})\times\left(\frac{\sum_{m'\in o}\exp(-\frac{\varepsilon U(G,m')}{2\Delta U})}{\sum_{m'\in o}\exp(-\frac{\varepsilon U(G,m')}{2\Delta U})}\right)\\
&=\exp(\varepsilon)\end{aligned}$$

It is clear that the process of selecting the threshold m satisfies differential privacy. Therefore, the SNEM algorithm satisfies differential privacy.

Theorem 2: The GMSM algorithm satisfies ε-differential privacy.

Proof: In the GMSM algorithm, the randomized response mechanism is used to modify the edges of each selected node, so that the process of edge modification satisfies ε-differential privacy.

Let R denotes a randomized response mechanism, $Pr[x\to y]$ represents the probability that $x \in \{0,1\}$ changes to $y \in \{0,1\}$. Given $q = 1 - p = e^{\varepsilon}/1+e^{\varepsilon}$, when $\varepsilon > 0$, so that $q > 1 - p$. For each node, a binary sequence is obtained according to the 2-hop subgraph of each node. Let two binary sequences $Se_1(e_1, e_2, \ldots, e_n)$ and $Se_2(e_{1'}, e_{2'}, \ldots, e_{n'})$ be neighbor sequences of one node. In addition, there is one different element between them. Without loss of generality, let $M(m_1, m_2, ..., m_n)$ be any output of R, the result is as follows,

$$\frac{p_r[R(S) = M]}{p_r[R(S') = M]} = \frac{p_r[s_1 \to m_1] \cdot \ldots \cdot p_r[s_n \to m_n]}{p_r[s'_1 \to m_1] \cdot \ldots \cdot p_r[s'_n \to m_n]}$$
$$= \frac{p_r[s_1 \to m_1]}{p_r[s'_1 \to m_1]} \le \frac{q}{p} = e^{\varepsilon}$$

Therefore, no matter Se_1 or Se_2 is input, the randomized response achieves the differential privacy.

After all nodes are modified by the randomized response, the shuffle model handles the results of the randomized response by using a shuffler. According to the post-processing, the GMSM algorithm satisfies ε-differential privacy.

Theorem 3: The UNDP algorithm satisfies ε-differential privacy.

Proof: In this algorithm, the Laplace Mechanism is used to added noise on edges.

Given two graphs G_a and G_b which are neighbors, the Hamming distance between G_a and G_b is the maximum degree of nodes in these two graphs. Let $F(.)$ be some identity mapping $F : G \to E$. so $F(G_a) \to E_a$, $F(G_b) \to E_b$, the sensitivity of F:

$$\Delta f = \max_{G_a, G_b} \left| F(G_a) - F(G_b) \right|_1$$
$$\Delta f = \left| f_1(G_a) - f_1(G_b) \right| + \left| f_2(G_a) - f_2(G_b) \right| + \ldots + \left| f_n(G_a) - f_n(G_b) \right|$$

Then according to the Laplace Mechanism, the Laplace noise is added to the output of F, where LM denotes the Laplace Mechanism, Ln represents the Laplace noise and Z is the result.

$$Z = LM(G_a) = F(G_a) + Ln$$
$$Z = LM(G_b) = F(G_b) + Ln$$
$$Z = \{z_1, z_2, \ldots, z_n\}$$

Let $Pn\ [Z(G_a)]$ represent the probability density function of $LM\ (G_a, F, \varepsilon)$, and $Pn\ [Z(G_b)]$ denotes the probability density function of $LM\ (G_b, F, \varepsilon)$, the proof is shown as follows. Therefore, the process of adding noise on edges satisfies differential privacy.

Then, according to the post-processing, the noised graph is converted to an uncertain graph, so the UNDP algorithm satisfies ε-differential privacy.

Theorem 4: The UGSM algorithm satisfies ε-differential privacy.

Proof: In this algorithm, the SNEM algorithm, the GMSM algorithm and the UNDP algorithm are utilized to generate an uncertain graph. In particular, these three algorithms all satisfy differential privacy. According to the parallel composition properties and Sequential Composition properties, it is evident that the UGSM algorithm satisfies ε-differential privacy.

$$
\begin{aligned}
&\frac{Pn[Z(G_a)]}{Pn[Z(G_b)]} = \frac{Pn[LM(G_a)]}{Pn[LM(G_b)]} = \frac{Pn[Ln(G_a)]}{Pn[Ln(G_b)]} \\
&= \frac{\frac{1}{2\frac{\Delta f}{\varepsilon}}\exp(-\frac{|Z-F(G_a)|}{\frac{\Delta f}{\varepsilon}})}{\frac{1}{2\frac{\Delta f}{\varepsilon}}\exp(-\frac{|Z-F(G_b)|}{\frac{\Delta f}{\varepsilon}})} = \frac{\exp(-\frac{|Z-F(G_a)|}{\frac{\Delta f}{\varepsilon}}}{\exp(-\frac{|Z-F(G_b)|}{\frac{\Delta f}{\varepsilon}})} \\
&= \frac{\exp(-\frac{\varepsilon|z_1 - f_1(G_a)|}{\Delta f})\ldots\exp(-\frac{\varepsilon|z_n - f_n(G_a)|}{\Delta f})}{\exp(-\frac{\varepsilon|z_1 - f_1(G_b)|}{\Delta f})\ldots\exp(-\frac{\varepsilon|z_n - f_n(G_b)|}{\Delta f})} \\
&= \prod_{i=1}^{n}\exp(\frac{\varepsilon|z_i - f_i(G_a)|}{\Delta f} - \frac{\varepsilon|z_i - f_i(G_b)|}{\Delta f}) \\
&= \prod_{i=1}^{n}\exp(\frac{\varepsilon(|z_i - f_i(G_a)| - |z_i - f_i(G_b)|)}{\Delta f}) \\
&\le \prod_{i=1}^{n}\exp(\frac{\varepsilon|f_i(G_a) - f_i(G_b)|}{\Delta f}) \\
&= \exp(\frac{\varepsilon(|f_1(G_a) - f_1(G_b)| + \ldots + |f_n(G_a) - f_n(G_b)|)}{\Delta f}) \\
&\le \exp(\frac{\varepsilon\Delta f}{\Delta f}) = e^{\varepsilon}
\end{aligned}
$$

5. ALGORITHM ANALYSIS

The proposed algorithm is evaluated in this section. First, the experiment data sets are introduced. Then, the proposed algorithm is analyzed in preserving privacy and data utility. Finally, the proposed algorithm is compared with other algorithms.

5.1 Data Sets

In our experiments, two kinds of experiment data are utilized, which include the synthetic data sets and the real data sets. The synthetic data sets are obtained from ER graphs, which contain 500 and 1000 nodes. The real data sets contain Facebook data with 4039 nodes and 63731 nodes, and Enron email network with 36692 nodes.

To evaluate the proposed algorithm, (k, ε_l)-obfuscation algorithm [8], Rand-Walk algorithm [31], UGDP algorithm [32] and LDPGen algorithm [19] are adopted for comparison. All simulation experiments run on an HP computer, which has Intel Core i5-8500 with 3.00GHz and 12GB memory. For programming, Python is used on the Microsoft Windows 7 operating system.

5.2 Privacy Evaluation

In this section, in order to evaluate the privacy preserving, we present the expectation of editing distance (*EED*) to test the uncertain algorithms.

5.2.1 Privacy measurement

When a graph is converted into an uncertain graph, there is a certain gap between them which can be measured by the editing distance. Because the edge in uncertain graphs is uncertain, the expectation of editing distance is introduced to measure the gap between an original graph and an uncertain graph. Moreover, it also can be used to evaluate preserving privacy.

The larger *EED*, the better privacy preserving.

It is well-known that the definition of edit distance between two deterministic graphs G_1, G_2 as follows,

$$D(G_1, G_2) = |E_1 \setminus E_2| + |E_2 \setminus E_1|$$

According to the formula above, the expected edit distance between the uncertain graph G'' and the deterministic graph G as follows,

$$EED[D(G,G'')] = \sum_{G_1'} P_r(G_1')D(G,G_1') = \sum_{e_i \in G}(1-P_i) + \sum_{e_i \notin G} P_i$$

where G_1' is sampled from G'', $Pr(G')$ indicates the probability of obtaining G_1' from the uncertain graph G''.

In UGSM algorithm, when an uncertain graph G_u is obtained, the expected edit distance between G_u and the graph G is,

$$EED[D(G, G_u)] = EED[D(G, G')] + EED[D(G', G_u)]$$

where G' is obtained by the GMSM algorithm, G_u is generated by the UNDP algorithm.

$$EED[D(G, G')] = e_k$$

where e_k equals the edit distance between two deterministic graphs G and G', which is calculated by the following formula,

$$e_k = |E_a| + |E_d|.$$

where $|E_a|$ denotes the number of edges which are added in G, where $|E_d|$ is the number of edges which are deleted from G.

Then there are no edges added and removed in the UNDP algorithm, thus, the expected edit distance between G_u and the graph G' is

$$EED\left[D(G', G_u)\right] = \sum_{e_i \in G_u}(1-P_i)$$

where e_i belongs to the edges set of G_u, p_i is the probability of the edge e_i.

The expectation of editing distance (*EED*) between G_u and the graph G is shown as follows,

$$EED\left[D(G,G_u)\right]=e_k+\sum_{e_i\in G_u}(1-P_i).$$

5.2.2 Privacy analysis

In order to evaluate the different uncertain graphs algorithm, we use the *EED* to measure the privacy preserving. The greater *EED*, the better privacy preserving this uncertain graph algorithm achieves. We execute all data sets 10 times by using our algorithm and other algorithms to average out the results.

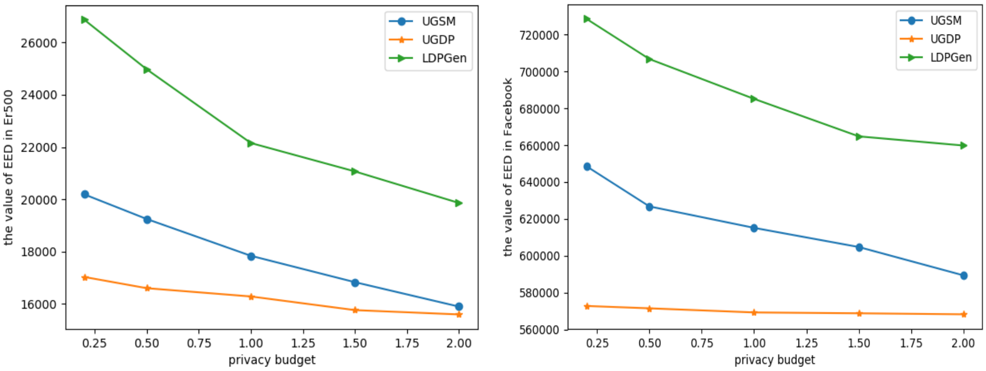
In the comparative experiments, the parameter of three algorithms is showed in Table 1. In (k, ε_l)-obfuscation algorithm, the obfuscation level k belongs to 10, 20, the tolerance parameter ε_l equals 0.1, the multiplier factor c is 1 and the white noise q is equal to 0.01. In Rand-Walk algorithm, the parameter t denotes the size of noise. In addition, the privacy budget ε in UGDP algorithm, LDPGen algorithm and UGSM algorithm is 0.2, 0.5, 1, 1.5, 2 ($\varepsilon = \varepsilon_1 = \varepsilon_2 = \varepsilon_3$).

Table 1. The *EED* values of five algorithms in different data sets.

Algorithm		ER 500	ER 1000	Facebook 4039	Enron 36692	Facebook 63731
UGSM	$\varepsilon = 0.2$	20187	73943	75864	304737	648567
	$\varepsilon = 0.5$	19243	71165	74132	298765	626787
	$\varepsilon = 1$	17634	70497	73654	283098	615143
	$\varepsilon = 1.5$	16823	68155	72234	276879	604735
	$\varepsilon = 2$	15654	67975	70925	266581	589347
(k, ε_l)-obfuscation	$k = 10$	13243	43512	48934	197865	457783
	$k = 20$	13654	43876	49263	198243	458495
Rand-Walk	$t = 5$	24754	81654	80432	357784	704356
	$t = 10$	24421	81243	79894	356465	703218
UGDP	$\varepsilon = 0.2$	17023	67889	62785	257863	572742
	$\varepsilon = 0.5$	16593	67254	61523	256890	571465
	$\varepsilon = 1$	16298	66865	60231	256135	569243
	$\varepsilon = 1.5$	15753	66734	59643	255764	568786
	$\varepsilon = 2$	15597	66452	59132	255365	568215
LDPGen	$\varepsilon = 0.2$	26876	83243	85832	374733	728577
	$\varepsilon = 0.5$	24975	81764	84231	358776	706786
	$\varepsilon = 1$	22154	78499	81654	333054	685156
	$\varepsilon = 1.5$	21065	77843	79239	316776	664798
	$\varepsilon = 2$	19853	76478	78092	308378	659823

The result of *EED* values are shown in Table 1. In Table 1, the *EED* values in the UGSM algorithm is shown from the first to the five rows. Moreover, the *EED* increases as the value of ε decreases, which means that the privacy preserving of the UGSM algorithm becomes stronger. For example, in FaceBook data set with 4039 nodes, when ε is 2, the value of *EED* is 70925. As ε ascends to 0.5, the value of *EED* rises to 75864, which means that the privacy preserving of UGSM algorithm is improved. Additionally, as the number of nodes in original graph increases, we can see that the *EED* of UGSM algorithm rises simultaneously, which indicates that the UGSM algorithm can provide privacy preserving for the different social networks. For instance, in Table1, when ε is 1, it is clear that the *EED* of UGSM algorithm increases from 17634 to 615143 as the number of nodes changes from 500 to 63731, which illustrates this algorithm can be applied in different social networks.

As shown in Table 1, the *EED* of (k, ε_l)-obfuscation algorithm is shown from the seventh row to the eighth row while the rows from the ninth to the tenth indicate the *EED* values of Rand-walk algorithm. In addition, the details of UGDP algorithm and LDPGen algorithm are described in the rest rows. Moreover, in the same data set, the value of *EED* obtained by UGSM algorithm is greater than that in (k, ε_l)-obfuscation algorithm and UGDP algorithm, but it is smaller than that in Rand-Walk algorithm and LDPGen algorithm. For example, in the FaceBook data set with 4039 nodes, when ε is 0.5, the value of *EED* in UGSM algorithm is 74132, while that in (k, ε_l)-obfuscation algorithm with $k = 10$ and UGDP algorithm are 48934 and 61523 respectively. Meanwhile, that in Rand-Walk algorithm with $t = 10$ is 79894 and that in LDPGen algorithm with the same ε is 84231. In particularity, the results show that the shuffle model applied in UGSM algorithm takes effect on the value of *EED*. Therefore, according to the definition of *EED*, it is clear that UGSM algorithm can provide stronger privacy preserving than (k, ε_l)-obfuscation algorithm and UGDP algorithm, but it is weaker than Rand-Walk algorithm and LDPGen algorithm.



(a) The comparison of *EED* in Er500. (b) The comparison of *EED* in Facebook.

Fig. 4. The comparison of algorithm in *EED*.

As shown in Fig. 4, where Fig. 4 (a) shows the values of *EED* in three differential privacy algorithms in the data set ER500, while Fig. 4 (b) gives the values of *EED* in three differential privacy algorithms in the data set Facebook (63731). In Fig. 4, the values of *EED* in UGSM algorithm, UGDP algorithm and LDPGen algorithm all raise as ε increases, which indicates that the larger ε, the better privacy preserving of three algorithms. Given a fixed ε, compared with UGDP algorithm, the value of *EED* in UGSM algorithm is greater, so the result shows the local differential privacy in this algorithm plays a role in privacy preserving. In addition, the value of *EED* in UGSM is smaller than that in LDPGen algorithm, which points out that UGSM algorithm has better data utility than LDPGen algorithm because the shuffle model can improve data utility. Thus, with the application of the shuffle model in UGSM algorithm, UGSM algorithm achieves the trade off between privacy preserving and data utility.

5.3 Utility Evaluation

5.3.1 Utility metrics

In order to evaluate the data utility, the *NE*, *AD* and *DV* are used in our experiments. Due to the uncertainty of edges in an uncertain graph, the degree of a node in an uncertain graph is the expected degree which is equal to the sum of probabilities of its adjacent edges. Therefore, the definitions the *NE*, *AD* and *DV* are shown as follows:

$$d_v = \sum p(i,j) \qquad NE = {}^{1}\!/_{2}\sum\nolimits_{v\in V} d_v$$

$$AD = {}^{1}\!/_{n}\sum\nolimits_{v\in V} d_v \qquad DV = {}^{1}\!/_{n}\sum\nolimits_{v\in V} (d_v - AD)^2$$

In addition, the diameter (SDiam) which denotes the maximum distance among all path-connected pairs of nodes is adopted. The second measure is the average distance (S_{APD}) which is the average shortest distance among all path-connected pairs of nodes.

Furthermore, we can measure data utility of each algorithm through *Utility* (function) defined as follows. Note that the greater the *Utility*, the better the data utility of this algorithm.

$$Utility = (1 - \frac{|UV - RV|}{RV}) \times 100\,\%$$

where *UV* is the graph metrics in uncertain graphs achieved by different algorithms, *RV* is the real metrics in the original graphs.

Finally, to compare the UGSM algorithm with other three algorithms in the data utility, we utilize the error on one graph metric which is described as follows,

$$\Delta q(G, G_u) = |q(G) - q(G_u)|$$

where q represents one graph metric.

5.3.2 Utility analysis

To evaluate the data utility of the uncertain graph algorithm, we get the experimental results by averaging the results 10 times and taking the final value. Table 2 illustrates the graph metrics in original graph and the UGSM algorithm.

As shown in Table 2, the value of *NE* in five data sets decreases as the ε rises, so does the value of *AD*. For instance, in the Facebook data set with 4039 nodes, the value of *NE* descends from 71648 to 69395 with the ε changing from 0.2 to 2, while the value of *AD* decreases from 35.65 to 33.76. In addition, the value of *DV* descends from 4567.76 to 3978.23, while the S_{APD} rises to 2.25. In the UNSM algorithm, the smaller ε, the more edges are modified in the original graph, so the greater the value of *NE* and *AD*.

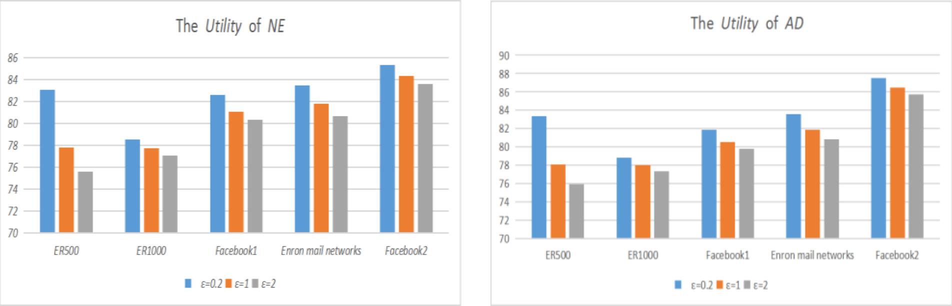
On the contrary, the larger ε, the fewer edges are modified, thus the value of DV becomes smaller and the S_{APD} is closer to that of original graph. Therefore, the UGSM algorithm can provide sufficient data utility by regulating the privacy budget ε.

Then we use the *Utility* to evaluate the data utility of UGSM algorithm. As shown in Fig. 5 (a), the maximum *Utility* of *NE* is 86%. In Fig. 5 (b), the highest *Utility* of *AD* can reach 88%, the lowest is 76%, so the average *Utility* of *AD* is about 82%. According to the results in Table 2, in the Facebook data set with 4039 nodes, the highest *Utility* of S_{Diam} is about 76%, while that of S_{APD} is 73%. Especially, the highest *Utility* of S_{APD} can reach 85% in the Facebook data set with 63731 nodes. Therefore, the data utility of UGSM algorithm is feasible.

In addition, we utilize the Δq to measure the comparison of data utility among UGS-M algorithm, (k, ε_l)-obfuscation algorithm, Rand-walk algorithm, UGDP algorithm and LDPGen algorithm. In the Facebook data set with 63731 nodes, the *NE* of the original graph is 817090, the *NE* obtained by UGSM algorithm is 686091 while the *NE* of the other four algorithms is 816286 ($k = 10$), 425702 ($t = 5$), 612833 ($\varepsilon = 0.2$) and 401253 ($\varepsilon = 0.2$) respectively. Thus, the value of the Δq of *NE* obtained by UGSM algorithm is larger than that in (k, ε_l)-obfuscation algorithm, but it is less than that in UGDP algorithm, the Rand-walk algorithm and LDPGen algorithm. The result indicates that UGSM algorithm

Table 2. The metrics in UGSM algorithm.

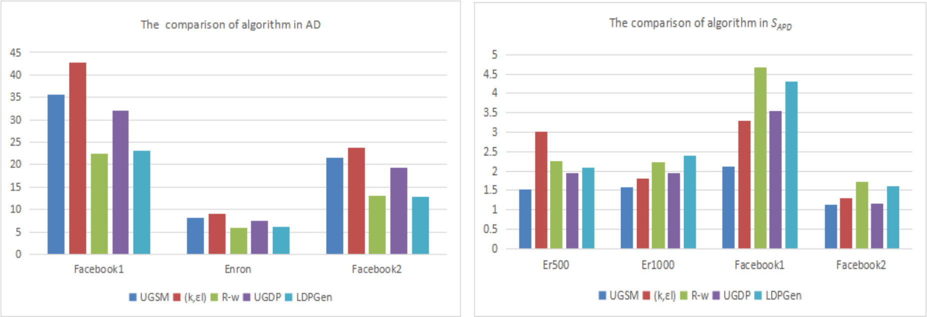
Data Sets	Metrics	Original Network	$\varepsilon = 0.2$	$\varepsilon = 1$	$\varepsilon = 2$
ER graph 500	NE	24844	19638	18534	17776
	AD	99	77.52	75.31	74.87
	DV	2607	3132.41	2971.28	2786.87
	S_{Diam}	4	2.34	2.46	2.63
	S_{APD}	1.80	1.53	1.67	1.74
ER graph 1000	NE	99902	77476	76649	75958
	AD	199	154.85	153.29	150.91
	DV	8376	9828.26	9542.69	9476.32
	S_{Diam}	4	2.46	2.78	2.86
	S_{APD}	1.80	1.57	1.70	1.93
Facebook 4039	NE	88234	71648	70732	69395
	AD	44	35.65	34.76	33.76
	DV	3262	4567.76	4132.89	3978.23
	S_{Diam}	4	2.76	2.92	3.10
	S_{APD}	3	2.12	2.23	2.25
Enron 36692	NE	183831	151457	149323	147032
	AD	10	8.21	8.12	7.93
	DV	1328	1963.34	1786.48	1623.32
	S_{Diam}	4	2.86	3.12	3.21
	S_{APD}	33.9	26.4	27.9	28.3
Facebook 63731	NE	817090	686091	672766	668876
	AD	25	21.65	21.52	21.37
	DV	1785	4134	3365	2956
	S_{Diam}	4	2.75	2.86	2.95
	S_{APD}	1.32	1.12	1.16	1.18



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

Fig. 5. The Utility of *NE* and *AD* in UGSM algorithm.

is not better than (k, ε_l)-obfuscation algorithm in the data utility, but it has better data utility than UGDP algorithm, the Rand-walk algorithm and LDPGen algorithm. Additionally, we describe the detail of the Δq of other graph metrics in Fig. 6, where Fig. 6 (a) shows the Δq about *AD* in different algorithms, while Fig. 6 (b) demonstrates the Δq about S_{APD}. According to the results, UGSM algorithm has better data utility than Rand-walk algorithm and LDPGen algorithm. In particular, UGSM algorithm is better than UGDP algorithm in some graph metrics, such as *NE* and *AD*.

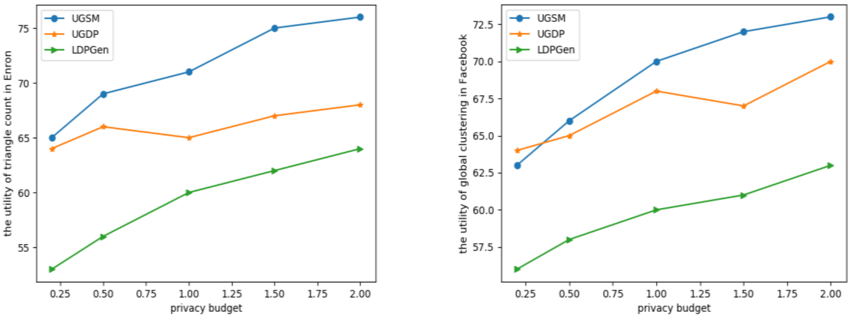


(a) The comparison of algorithms in *AD*. (b) The comparison of algorithms in S_{APD}.

Fig. 6. The comparison of different algorithms.

Furthermore, as shown in Fig. 7, where Fig. 7 (a) describes the *Utility* of the triangle count in UGSM algorithm, UGDP algorithm and LDPGen algorithm in the data set Enron, while Fig. 7 (b) shows the *Utility* of the global clustering coefficient in these three algorithms in the data set Facebook (63731). In Fig. 7 (a), the *Utility* of the triangle count in UGSM algorithm increases as ε rises, so does LDPGen algorithm. However, the *Utility* of the triangle count in UGDP algorithm changes little with ε increasing. In Fig. 7 (b), the change trend of the global clustering coefficient in three algorithms is similar to that of the triangle count. According to the results in Fig. 7, the data utility of UGSM algorithm between UGDP algorithm and LDPGen algorithm, which illustrates the shuffle model can improve the data utility of UGSM algorithm.

In summary, the performance of experiments shows that the UGSM algorithm can not only provide sufficient privacy preserving, but also maintain data utility.



(a) The comparison of triangle count in *Enron*.

(b) The comparison of clustering coefficient in *Facebook*.

Fig. 7. The comparison of utility in different algorithms.

6. CONCLUSION

The rapid development of mobile wireless technology facilitates the popularization of mobile social networks, which makes our daily life more and more convenient. However, the individual privacy problem in the mobile social network has become an urgent problem because a large amount of data containing individual privacy information is collected to the mobile social network. In order to solve this problem, many methods have been proposed, including graph modification methods, differential privacy based methods. Recently, although differential privacy methods have been widely used for graph data, how to realize the effective protection of differential privacy to graph data is a very urgent problem

To solve the problem, we combine shuffle model with the uncertainty graph method to protect link privacy in social networks. In this method, the shuffling model realizes the edge modification of the original graph, which can not only protect the sensitive relationships of nodes, but also improve the insufficient utility of LDP. At the same time, the uncertain method preserves the sensitive relationships of nodes by injecting uncertainty on edges while maintaining the structure of original graph. In addition, the exponential mechanism is utilized to select nodes, which reduces the interference caused by differential privacy and effectively improves the data utility. The theoretical analysis shows that the proposed method satisfies the differential privacy. In addition, the results of experiments demonstrate that the proposed method can effectively provide strict privacy guarantee and maintain data utility.

In the future, as this method achieves a better balance between privacy and utility, how to apply shuffle model to complex networks, such as distributed networks and directed networks, is our next work.

ACKNOWLEDGMENTS

This research was funded by the National Natural Science Foundation of China (62001273, 61962033) the Fundamental Research Funds for the Central Universities (No. 2019CBLY004 and No. GK201704017), the Scientific and Technological Project of Shangluo (No.2021-C-0004) and Shangluo Universities Key Disciplines Project, Discipline name: Mathematic.

REFERENCES

1. C. Sandeepa, B. Siniarski, N. Kourtellis, S. Wang, and M. Liyanage, “A survey on privacy for B5G/6G: New privacy challenges, and research directions,” *Journal of Industrial Information Integration*, Vol. 30, 2022, pp. 1-37.
2. A. Majeed, S. Khan, and S. O. Hwang, “A comprehensive analysis of privacy-preserving solutions developed for online social networks,” *Electronics*, Vol. 11, 2022, pp. 1-37.
3. L. Yin, J. Feng, H. Xun, Z. Sun, and X. Cheng, “A privacy-preserving federated learning for multiparty data sharing in social IoTs,” *IEEE Transactions on Network Science and Engineering*, Vol. 8, 2021, pp. 2706-2718.
4. S. M. Safi, A. Movaghar, and M. Ghorbani, “Privacy protection scheme for mobile social network,” *Journal of King Saud University – Computer and Information Sciences*, Vol. 34, 2022, pp. 4062-4074.
5. J. Isaak and M. J. Hanna, “User data privacy: Facebook, Cambridge analytica, and privacy protection,” *Computer*, Vol. 51, 2018, pp. 56-59.
6. J. Shen, J. Tian, Z. Wang, and H. Cai, “Friendship links-based privacy-preserving algorithm against inference attacks,” *Journal of King Saud University – Computer and Information Sciences*, Vol. 34, 2022, pp. 9363-9375.
7. K. R. Langari, S. Sardar, and A. A. S. Mousavi, “Combined fuzzy clustering and firefly algorithm for privacy preserving in social networks,” *Expert Systems with Applications*, Vol. 141, 2020, pp. 1-12.
8. P. Boldi, F. Bonchi, A. Gionis, and T. Tassa, “Injecting uncertainty in graphs for identity obfuscation,” in *Proceedings of the VLDB Endowment*, Vol. 5, 2012, pp. 1376-1387.
9. X. Ding, C. Wang, K. K. R. Choo, and H. Jin, “A novel privacy preserving framework for large scale graph data publishing,” *IEEE Transactions on Knowledge and Data Engineering*, Vol. 33, 2021, pp. 331-343.
10. C. Dwork, “Differential privacy,” in *Proceedings of the 33rd International Colloquiium on Automata, Languages and Programming*, 2006, pp. 1-12.
11. H. Jiang, J. Pei, D.Yu, J. Yu, B. Gong, and X. Cheng, “Applications of differential privacy in social network analysis: a survey,” *IEEE Transactions on Knowledge and Data Engineering*, Vol. 35, 2021, pp. 108-127.
12. L. Hou, W. Ni, S. Zhang, N. Fu, and D. Zhang, “Block-HRG: Block-based differentially private IoT networks release,” *Ad Hoc Networks*, Vol. 140, 2023, pp. 1-12.
13. S. Lan, H. Xin, W. Yingjie, and G. Yongy, “Sensitivity reduction of degree histogram publication under node differential privacy via mean filtering,” *Concurrency and Computation: Practice and Experience*, Vol. 33, 2021, pp. 1-8.
14. F. Ahmed, A. X. Liu, and R. Jin, “Publishing social network graph eigenspectrum with privacy guarantees,” *IEEE Transactions on Network Science and Engineering*, Vol. 7, 2020, pp. 892-906.
15. J. Laeuchli, Y. Ramírez-Cruz, and R. Trujillo-Rasua, “Analysis of centrality measures under differential privacy models,” *Applied Mathematics and Computation*, Vol. 412, 2022, pp. 1-11.

16. T. Gao and F. Li, "Sharing social networks using a novel differentially private graph model," in *Proceedings of the 16th IEEE Annual Consumer Communications and Networking Conference*, 2019, pp. 1-4.
17. F. Z. Errounda and Y. Liu, "Collective location statistics release with local differential privacy," *Future Generation Computer Systems*, Vol. 124, 2021, pp. 174-186.
18. L. Lyu, H. Yu, X. Ma, C. Chen, L. Sun, J. Zhao, and S. Y. Philip, "Privacy and robustness in federated learning: Attacks and defenses," *IEEE Transactions on Neural Networks and Learning Systems*, 2022, pp. 1-21.
19. 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 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 425-438.
20. C. Wei, S. Ji, C. Liu, W. Chen, and T. Wang, "ASGLDP: Collecting and generating decentralized attributed graphs with local differential privacy," *IEEE Transactions on Information Forensics and Security*, Vol. 15, 2020, pp. 3239-3254.
21. P. Liu, Y. Xu, Q. Jiang, Y. Tang, Y. Guo, L. E. Wang, and X. Li, "Local differential privacy for social network publishing," *Neurocomputing*, Vol. 391, 2020, pp. 273-279.
22. H. Huang, Y. Yang, and Y. Li, "Local privacy preserving synthetic social graph generation," in *Proceedings of International Conference on Collaborative Computing*: *Networking, Applications and Worksharing*, 2021, pp. 389-404.
23. T. Wang, B. Ding, M. Xu, Z. Huang, C. Hong, J. Zhou, and S. Jha, "Improving utility and security of the shuffler-based differential privacy," *arXiv Preprint*, 2020, arXiv: 1908.11515, pp. 1-15.
24. A. Girgis, D. Data, S. Diggavi, P. Kairouz, and A. T. Suresh, "Shuffled model of differential privacy in federated learning," in *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, 2021, pp. 2521-2529.
25. J. Imola, T. Murakami, and K. Chaudhuri, "Differentially private subgraph counting in the shuffle model," *arXiv Preprint*, 2022, arXiv:2205.01429, pp. 1-25.
26. R. Mortazavi and S. H. Erfani, "GRAM: An efficient (*k*, *l*) graph anonymization method," *Expert Systems with Applications*, Vol. 153, 2020, pp. 1-9.
27. H. Zhang, L. Lin, L. Xu and X. Wang, "Graph partition based privacy-preserving scheme in social networks," *Journal of Network and Computer Applications*, Vol. 195, 2021, pp. 1-12.
28. M. Kiabod, M. N. Dehkordi, and B. Barekatain, "A fast graph modification method for social network anonymization," *Expert Systems with Applications*, Vol. 180, 2021, pp. 1-19.
29. 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 of Things Journal*, Vol. 9, 2021, pp. 2761-2777.
30. N. Fu, W. Ni, S. Zhang, L. Hou, and D. Zhang, "GC-NLDP: A graph clustering algorithm with local differential privacy," *Computers* & *Security*, Vol. 124, 2023, pp. 1-14.
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. Wu, H. Liu, and Y. H. Zhou, "A privacy-preserving approach in friendly-correlations of graph based on edge-differential privacy," *Journal of Information Science and Engineering*, Vol. 35, 2019, pp. 821-837.
33. J. Yan, Y. Tian, H. Liu, and Z. Wu, "Uncertain graph generating approach based on differential privacy for preserving link relationship of social networks," *International Journal of Security and Networks*, Vol. 17, 2022, pp. 28-38.
34. T. Lv, H. Li, Z. Tang, F. Fu, J. Cao, and J. Zhang, "Publishing triangle counting histogram in social networks based on differential privacy," *Security and Communication Networks*, Vol. 2021, 2021, pp. 1-16.
35. M. Iftikhar, Q. Wang, and Y. Li, "dK-Personalization: publishing network statistics with personalized differential privacy," in *Proceedings of Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 2022, pp. 194-207.
36. M. Eliáš, M. Kapralov, J. Kulkarni, and Y. T. Lee, "Differentially private release of synthetic graphs," in *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2020, pp. 560-578.
37. S. Zhang, W. Ni, and N. Fu, "Community preserved social graph publishing with node differential privacy," in *Proceedings of International Conference on Data Mining*, 2021, pp. 1400-1405.
38. M. Yang, L. Lyu, J. Zhao, T. Zhu, and K. Y. Lam, "Local differential privacy and its applications: A comprehensive survey," *arXiv Preprint*, 2020, arXiv:2008.03686, pp. 1-25.
39. J. Imola, T. Murakami, and K. Chaudhuri, "Locally differentially private analysis of graph statistics," in *Proceedings of the 30th USENIX Security Symposium*, 2021, pp. 983-1000.
40. 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 Transactions on Knowledge and Data Engineering*, Vol. 34, 2020, pp. 4905-4920.
41. J. Yang, X. Ma, X. Bai, and L. Cui, "Graph publishing with local differential privacy for hierarchical social networks," in *Proceedings of the 10th International Conference on Electronics Information and Emergency Communication*, 2020, pp. 123-126.
42. V. Feldman, A. McMillan, and K. Talwar, "Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling," in *Proceedings of IEEE 62nd Annual Symposium on Foundations of Computer Science*, 2021, pp. 954-964.
43. S. Biswas, K. Jung, and C. Palamidessi, "Tight differential privacy guarantees for the shuffle model with *k*-randomized response," *arXiv Preprint*, 2022, arXiv:2205. 08858, pp.1-13.
44. A. Cheu, A. Smith, J. Ullman, D. Zeber, and M. Zhilyaev, "Distributed differential privacy via shuffling," in *Proceedings of International Conference on Theory and Applications of Cryptographic Techniques*, 2019, pp. 375-403.
45. S. P. Liew, T. Takahashi, S. Takagi, F. Kato, Y. Cao, and M. Yoshikawa, "Network shuffling: Privacy amplification via random walks," *arXiv Preprint*, 2022, arXiv: 2204.03919, pp. 1-15.

Jun Yan (顏軍) received the MS 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.

Wen-Li Wang (王文麗) is a Postgraduate of Shaanxi Normal University. Her main research interest is privacy preserving.

Zhen-Qiang Wu (吳振強) received his BS degree in 1991 from Shaanxi Normal University, China, and received his MS 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.

Lai-Feng Lu (魯來風) received MS 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 preserving.

Yi-Hui Zhou (周異輝) received her BE, MS and Ph.D. degrees 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.