Privacy Protection Method for Sensitive Weighted Edges in Social Networks

Weihua Gong[1], Rong Jin[2*], Yanjun Li[1*], Lianghuai Yang[1], and Jianping Mei[1]
[1] School of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China
[e-mail: {whgong, yjli, yanglh, jpmei}@zjut.edu.cn]
[2] School of Informatics and Electronics, Zhejiang Sci-Tech University, Hangzhou 310018, China
[e-mail: jr@zstu.edu.cn]
*Corresponding author: Yanjun Li, Rong Jin

Received December 23, 2019; accepted August 31, 2020;
published February 28, 2021

Abstract

Privacy vulnerability of social networks is one of the major concerns for social science research and business analysis. Most existing studies which mainly focus on un-weighted network graph, have designed various privacy models similar to k-anonymity to prevent data disclosure of vertex attributes or relationships, but they may be suffered from serious problems of huge information loss and significant modification of key properties of the network structure. Furthermore, there still lacks further considerations of privacy protection for important sensitive edges in weighted social networks. To address this problem, this paper proposes a privacy preserving method to protect sensitive weighted edges. Firstly, the sensitive edges are differentiated from weighted edges according to the edge betweenness centrality, which evaluates the importance of entities in social network. Then, the perturbation operations are used to preserve the privacy of weighted social network by adding some pseudo-edges or modifying specific edge weights, so that the bottleneck problem of information flow can be well resolved in key area of the social network. Experimental results show that the proposed method can not only effectively preserve the sensitive edges with lower computation cost, but also maintain the stability of the network structures. Further, the capability of defending against malicious attacks to important sensitive edges has been greatly improved.

Keywords: Social Network, Privacy Protection, Sensitive Weighted Edges, Edge Betweenness

This research was supported by National Natural Science Foundation of China under grant Nos. 61772472, 61070042, Natural Science Foundation of Zhejiang Province under grant Nos. LY13F020026, LY20F020023, the China Postdoctoral Science Foundation under grant No. 2015M581957, and the Fundamental Research Funds for the Provincial Universities of Zhejiang under grant No. RF-A2019002.

http://doi.org/10.3837/tiis.2021.02.009 **ISSN : 1976-7277**

1. Introduction

In recent years, the growing popularity of social networks has profoundly transformed how people live, work and communicate. Nowadays a large number of famous social applications such as Facebook, Twitter and Weibo have gathered numerous users. In these apps, active users have intimately connected not only the cyber social space but also the physical society in real world which is consolidated into one whole entity, generating great commercial value and social significance. Unfortunately, people chronically rely on these social applications while ignoring the privacy protection of their important information such as users' attributes and relationships in reality. Today various privacy attacks and disclosures of social networks have triggered a series of serious security threats and social anxiety issues. Therefore, it seems particularly urgent to study the privacy protection technology of social networks.

As known to all, social networks being affiliated to the field of complex networks science, not only contain vertex attribute data but also include their relational data. Various privacy protection methods have been proposed aiming at protecting these two types of privacy objects. For example, regarding the privacy of vertex attributes, data generalization [1-4] or perturbation methods [5-7] are usually adopted to protect personal identity or sensitive attributes such as name, phone number, address, etc from disclosure. On the other hand, for the privacy of the relational data [8], it has already become one of the research hotspots that needs to be explored more intensively. In order to achieve this privacy goal, social networks are usually modeled as graph structures and then edge perturbation [9] or graph modification [10] methods are used to modify edge weights, or randomly adding, deleting vertices or edges. In addition, there also exists other anonymizing methods to preserve the privacy of weighted network graphs by removing sensitive edges or edge clustering anonymization [11]. On the whole, most of the existing studies on privacy protection mainly tend to focus on how to implement different anonymization models based on vertex attributes or relation graphs, while ignoring the huge information loss and the calculation of NP-hard problems due to anonymization. Moreover, they all have the deficiencies of the social relationships in network structures being seriously damaged. Thus, current results show that we need further discussions for preserving the privacy of sensitive weighted edges.

In general, the main contributions of this paper are summarized as follows.

1) This paper formalizes the notion of sensitive edges considered as privacy data, which are differentiated from the normal weighted edges in the social network graph. We refer the sensitive edges as hubs or betweenness centrality with high betweenness values that exist on many shortest paths throughout graph.

2) We propose a novel graph reconstruction technique based on perturbation operations to preserve the privacy of sensitive weighted edges, by means of adding new perturbed edges or modifying the weights of sensitive edges in social network graph.

3) We provide the metric of information loss to evaluate the cost of privacy preservation, and present the results of experiments on three public datasets to prove that our proposed perturbation method can not only preserve the structural properties of network graph with lower cost, but also improve the capability of defending against malicious attacks to important sensitive edges.

The rest of this paper is organized as follows. Section 2 briefly reviews the related work in privacy protection of social network. Section 3 formally defines preliminary concepts and notations. Section 4 proposes perturbation-based privacy protection algorithms for sensitive weighted edges in network graph. Section 5 presents experimental results using various evaluation metrics. Finally, Section 6 concludes this paper.

2. Related Work

In the past decade, privacy issues in social networks have been extensively studied. The surveys in [12, 13] have summarized recent existing privacy protection techniques of graph structures in social networks. Generally speaking, existing privacy protection approaches for vertices' attributes and relational data in network graphs can be categorized into two groups: data perturbation [5-7, 9, 10] and *k*-anonymity [14-16].

We note that as one typical clustering based model, the basic idea of *k*-anonymity is to make the probability of re-identification attack with background knowledge no larger than 1/*k* by way of clustering the vertices or generalizing the structures in network graph. Some researchers including [14-16] have proposed to cluster *k* vertices with similar attributes or links into one super class and then generalize them to protect the privacy vertices, since their clustering objects are particularly targeted at the network vertices, which leads to the serious defect that huge information loss of vertices will seriously incur graphical structure uncertainty. Besides previous *k*-anonymity model, other similar anonymization techniques are proposed in succession. For instance, Zou et al. [17] have proposed a kind of *k*-automorphism protection model to defend against structural attacks, this model is based on graph isomorphism theory to achieve privacy protection by adding and deleting edges. Cheng et al. [18] also design one similar *k*-isomorphism model to protect both vertices and links: a graph is *k*-isomorphism if this graph consists *k* disjoint isomorphic subgraphs. In addition, there are some studies in [19-21] aiming at the privacy issues of weighted edges in networks. Among them, Liu and Yang [19] propose the *k*-possible anonymity for a weighted graph based on edge generalization approach, the so-called *k*-possible graph refers to graph anonymization that vertices in the same anonymization group are indistinguishable from each other based on weight bags, and the adversary cannot re-identify each vertex with confidence larger than 1/*k*. Moreover, a novel *k*-weighted degree anonymous model is proposed in literature [20], this model makes sure for any vertex *u*, there are at least *k*-1 other vertices which have the same degree as *u* and the weights on the edges adjacent to these vertices are also the same as *u*, which helps to prevent vertex re-identification in the weighted graph based no distance functions. Likewise, another *k*-histogram anonymization proposed in literature [21] makes the weight bags of at least *k*-1 other weight bags that are same weight so as to prevent from weighted-based attacking.

With regarding to data perturbation approach, as is known that the original network graph may be randomly modified by adding or deleting edges or vertices, so that the attacker can't accurately conjecture the real structures. Intuitively, this approach can be further classified as graph structural perturbation and weight value perturbation. For instance, Hay et al. [14] develop a random graph perturbation method by randomly deleting or adding edges to anonymize a social network, which can effectively reduce the re-identification attacks by an adversary with acceptable distortion of the graph. In order to defend against vertex re-identification, Ying and Wu [9] propose spectrum preserving randomization method that belongs to edge based graph perturbation through removing true edges and adding some fake edges to maintain about the same number of edges before and after anonymization. Furthermore, Xiao et al. [22] extract a dendrogram from a simple graph according to hierarchical random graph (HRG) model, and then apply differential privacy methods to add noise so as to perturb the graph structure. On the other hand, as to edge weight perturbation, Liu et al. [23] use Gaussian randomization multiplication to modify the weight of specific edges in the meanwhile to keep the shortest paths between the specific pairs of vertices unchanged. Das et al. [24] consider edge weight anonymization in social graphs, and propose linear programming method to perturb the edge weights in the anonymized graph. As can be

seen, the edge weight perturbation method is only suitable to distort the original data values for one-dimensional distribution, consequently that will affect the data utility and privacy-preserving level. Unfortunately, until now there are only a few studies such as [10, 11, 25, 26] have discussed several defensive methods against inference attacks to network sensitive edges. Specifically, the approach of edge based graph randomization is proposed to protect sensitive links via adding or deleting operations in literature [10], but its disadvantage is that the original network structure will be seriously damaged. Then, another graph anonymization technique by removing sensitive edges or edge clustering anonymization is presented to preserve graph-based privacy attacks such as link re-identification in literature [11], whereas the data utility of privacy is determined by the amount of data removed. In addition, Yuan et al. [25] combine *k*-degree anonymity with *l*-diversity model that considers the protection of structural information as well as sensitive labels of individuals, which realizes the graph anonymization by adding noise vertices into a graph with the least distortion to the properties of the original graph, such as degrees and distances between vertices. However, this model does not consider the impact of the amount of information loss caused by graph distortion. Liu et al. [26] develop a general framework for preventing link inference attacks, which adopts a novel lineage tracking mechanism by edge-cutting, adding or switching operations to cut off the inference paths of sensitive relationships meanwhile retaining the data utility.

Overall, most previous privacy protection studies in social networks mainly focus on *k*-anonymity or data perturbation methods, but they may suffer from the serious problems of huge information loss and significant modification of key properties to network structure. Consequently, more attention still needs to be paid to preserve the privacy of important and sensitive parts in social networks. So far, many researchers have not yet been able to provide uniform definition about sensitive edges of network graph, especially for weighted edges not being emphasized enough. To overcome these limitations, this paper proposes a privacy protection method for weighted sensitive edges in social network graph. Our method firstly distinguishes sensitive edges and non-sensitive edges according to edge betweenness features of graph, and then uses the combination operations by adding noise edges and perturbating weight values for important sensitive weighted edges, to further preserve the privacy of key areas in social networks.

3. Preliminaries and Modeling

To be convenient, we firstly consider the social networks as undirected and weighted network graphs denoted by $G=(V,E)$, in which the set of vertices is $V=\{v_i \mid 1\le i\le n\}$, and the set of weighted edges is $E=\{e_{xy} \mid v_x, v_y \in V\}$, where the larger the weight of edge e_{xy} is, the closer the relationship between the two vertices v_x and v_y. In weighted network graph, the characteristic of edge betweenness reflects the necessary path through which the information flow or communication between vertices must pass in the whole graph. Note that the higher the edge betweenness value is, the greater importance to social network, so if the attackers deliberately aim at destroying the privacy of this area, which will cause more serious damages to the network structure than other parts. Thus, we formally define the edge betweenness as follows.

Definition 1. Edge betweenness: The ratio of the number of the shortest paths passing through edge e_{xy} to the total number of shortest paths among all pairs of vertices in network graph is called betweenness of edge e_{xy}, that can be expressed as:

$$\mathrm{BC}(e_{xy}) = \sum_{s \neq t} \frac{m_{st}^{e_{xy}}}{M} \tag{1}$$

where M represents the total number of shortest paths between all vertex pairs in network graph, and $m_{st}^{e_{xy}}$ is the number of the shortest paths that go through edge e_{xy} for two arbitrary vertices supposing v_s and v_t.

Next, the notion of the shortest path generally means the minimum length or hops from the source vertex to the sink vertex, but we note that there usually exist not only one path between the source and the sink vertices. Thus, in this paper we introduce another specific definition for shortest path.

Definition 2. Shortest path: Let $P_k(v_i, v_j) = v_i\ e_{i.} \ldots e_{.j}\ v_j$ be the k^{th} path between arbitrary vertex pair such as between v_i and v_j in network graph G, the sum of edge weights on the path is called the path weight $W_{P_k(v_i,v_j)} = \sum_{e_{i.} \in P_k(v_i,v_j)} e_{i.}$. Then, if the path has the smallest path weight among all the path sets P from vertex v_i to v_j is called the shortest path $P^*(v_i, v_j)$, that is, the condition must be satisfied:

$$W_{P^*(v_i,v_j)} = \min_{P}\{W_{P_k(v_i,v_j)}\} = \min_{P}\{\sum_{e_{i.} \in P_k(v_i,v_j)} e_{i.}\} \tag{2}$$

Here, notation $e_{i.}$ means the minimal edge weight on paths from vertex v_i to its neighbor.

As we can see from above, edge betweenness is closely related to all the weighted paths of vertex pairs in graph, where the edge with higher betweenness rather than higher weight maybe have greater influence to the whole network. Thus in this paper, our main goal is to how preserve the privacy of the most valuable or important edges in network graph. To determine which edges are valuable and need to be protected, we will distinguish the sensitive and non-sensitive edges based on the threshold value of edge betweenness.

Definition 3. Sensitive edge: If and only if the betweenness value of edge e_{xy} is no less than the threshold λ (λ >0), then the edge e_{xy} is said to be a sensitive edge, and denoted as e^*_{xy}.

$$e_{xy} \rightarrow e^*_{xy} \text{ iff } \mathrm{BC}(e_{xy}) \geq \lambda \tag{3}$$

According to above criterion, all edges in network graph will be grouped into two categories, sensitive edges and non-sensitive edges. Note that sensitive edges are obviously more valuable and influential than other non-sensitive edges, and such edges should be considered as privacy objects to be preserved. This paper will adopt perturbation operations to guarantee the privacy of sensitive edges, mainly including two types of operations such as adding new weight edges and modifying the weights of sensitive edges. As known that the graph perturbation will change the topological structure of a graph and along with the rapid reduction of data utility. In order to estimate the cost of graph perturbation, we introduce the edge-based metric for information loss that quantifies the total weights modified of the number of changed edges before and after graph perturbation.

Definition 4. Information loss: The sum of the edge weight changed occurring between the perturbed network G' and the original network G is considered as information loss.

$$\mathrm{IL}(G'/G)=\sum_{i=1}^{n}e_i^{'}+\sum_{j=1}^{m}|e_j\text{-}e_j^{'}| \tag{4}$$

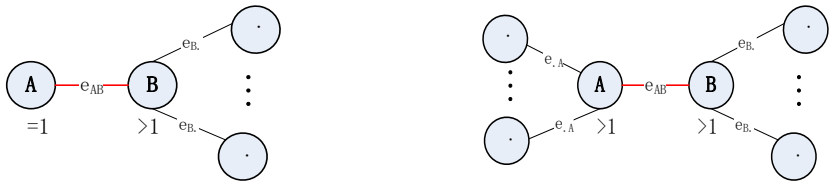
where the first term represents the sum of the weights for all added edges, and the second term means the accumulative sum of the changed weight for all modified edges. Intuitively, we can see that the less information loss manifests the smaller impact on data utility after perturbation.

4. Privacy Preservation Algorithm for Sensitive Weighted Edges

As is pointed out, sensitive edges and their neighbors are becoming the most important properties of social networks. In order to guarantee the privacy of these key areas, this paper will adopt graph perturbation approach so as to reduce the betweenness of sensitive edges. In this section, we will give the structural analysis of each sensitive edge and its neighbors in network graph, and then our perturbation operations are proposed for the different cases of sensitive edges.

4.1 Structure Analysis of Neighbors for Sensitive Edges

Assuming arbitrary sensitive edge in network graph, there exist two cases for neighboring edges on both sides of such edge as shown in **Fig. 1**.



(a) (b)

Fig. 1. Neighbor edges on both sides of sensitive edges

In the case shown in **Fig. 1(a)**, either side of the sensitive edge e_{AB} has more than one neighbors, for example, the degree of the vertex v_B in edge e_{AB} is greater than one, while the other side of such edge for instance vertex v_A has only one degree. Thus, in this case the sensitive edge e_{AB} satisfies the below condition.

$$e_{AB}\in\left\{e_{\mathrm{xy}}\,|\,v_x\times\left(V-v_x-v_y\right)\neq\varnothing\wedge v_y\times\left(V-v_x-v_y\right)=\varnothing\right\}\subset E \tag{5}$$

From the view of network structure, we know that this type of sensitive edge e_{AB} lies in the border position of social network. Because there is only one side of this sensitive edge connected with multiple neighbors, and thus such edge e_{AB} becomes an important area in network graph. Therefore, it is of great significance to protect the privacy of important endpoint of sensitive edge, for example vertex v_B and its neighbors.

In the case shown in **Fig. 1(b)**, both sides of the sensitive edge e_{AB} have multiple neighbors, and the degrees of these vertices for example v_A and v_B are both greater than one. In this situation, both sides of such sensitive edge e_{AB} should satisfy the below condition.

$$e_{AB} \in \left\{ e_{xy} \mid v_x \times \left(V - v_x - v_y \right) \neq \varnothing \wedge v_y \times \left(V - v_x - v_y \right) \neq \varnothing \right\} \subset E \tag{6}$$

To consider this case, the sensitive edge e_{AB} actually lies in the critical positions where will be always passed by many shortest paths between vertex pairs in network graph, which means the sensitive sub-paths. Thus, how to perturb the both sides of this important sensitive edge is the most important and complex tasks.

4.2 Perturbation Operations for Sensitive Weighted Edges

Firstly, for the case shown in **Fig. 1(a)**, since this type of sensitive edge is just on the marginal position of the social network, we only need to consider the perturbation operation to one side of such sensitive edge with vertex degree greater than one. In this situation, we perturb the graph by adding a new weighted edge same to the sensitive edge e_{AB}, which connects from one side of the sensitive edge with no neighbors to the nearest neighbor of the other side, and the nearest neighbor comes from the adjacent set of this sensitive edge. **Fig. 2** shows the operation results of graph perturbation, assume the sensitive edge is e_{AB}, we select the nearest neighbor with minimum weight from the adjacent set of vertex v_B in sensitive edge, for instance v_x, this can be expressed as:

$$Min\{e_{Bx}\}, s.t. v_x \in N(v_B) \tag{7}$$

where the notation $N(v_B)$ means the adjacent set of vertex v_B. Then, we let the vertex v_x connect to the vertex v_A of the sensitive edge and copy the same weight with $e_{Ax} = e_{AB}$, thus obviously the shortest path $P^*(v_A, v_x)$ from vertex v_x to vertex v_A will be replaced by the new path v_A, e_{AB}, v_x.

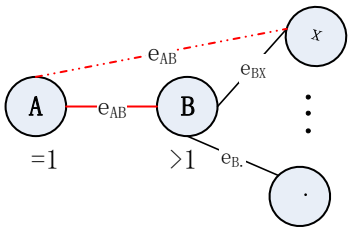


Fig. 2. Perturbation by adding new edge with the same weight

Secondly, for the case shown in **Fig. 1(b)**, note that this type of sensitive edge e_{AB} usually belongs to the very important area in social network, we will consider perturbation operations for the adjacent vertices at both ends of the sensitive edge. To discuss all possible situations of the neighboring vertices adjacent to the sensitive edge, we further group the relationship types of neighboring vertices into three different situations. That depending on whether the neighboring vertices in the two ends of the sensitive edge e_{AB} are either connected to each other or not, or two ends of such sensitive edge have the same shared neighboring vertex, more detailed cases are demonstrated in **Fig. 3**.

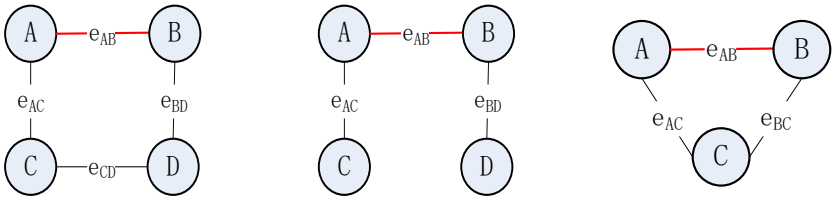


Fig. 3. Relation types between the sensitive edge and neighboring vertices

For the situation in **Fig. 3(a)**, when the two neighboring vertices at both sides of the sensitive edge e_{AB} are connected with the weight e_{CD}, we may directly perturb the weight of the connected neighbors by changing the old weight to the new sensitive weight e_{AB}, the result is shown in **Fig. 4(a)**. After the weight being perturbed, we note that all the shortest paths that contain the sub-paths from one neighbor vertex v_C to the other neighbor vertex v_D denoted by $P(v_C, v_D) = v_C, e_{CA}, v_A, e_{AB}, v_B, e_{BD}, v_D$ which passed through the sensitive edge e_{AB}, will be directly changed to pass through the more shorter sub-path v_C, e_{AB}, v_D, thereby the edge betweenness of the original sensitive edge e_{AB} will correspondently fall down by modifying the weight of next adjacent edge.

For the situation in **Fig. 3(b)**, when the neighboring vertices at both sides of the sensitive edge e_{AB} are not directly connected, we should take perturbation operation by adding new pseudo edge between the two neighboring vertices with the same weight to the sensitive edge, and the corresponding operation is as shown in **Fig. 4(b)**. More specifically from the graph, the disconnected neighboring vertices v_C and v_D are added a new pseudo-edge between them and given the same sensitive weight e_{AB}. Similarly, we can conclude that the previous indirect shortest path $P(v_C, v_D)$ will be turned into the direct shortest path v_C, e_{AB}, v_D, which will also lead to the betweenness of original sensitive edges e_{AB} decreased slightly.

For the situation in **Fig. 3(c)**, there is a common neighboring vertex for the sensitive edge e_{AB}, and if the condition is satisfied by $e_{AB} + e_{BC} < e_{AC}$, then we should change the larger weight of edge e_{AC} into the sensitive weight e_{AB}, so that makes the shortest path between vertices v_A and v_C is directly changed to v_A, e_{AB}, v_C. Consequently, the result is shown in **Fig. 4(c)**.

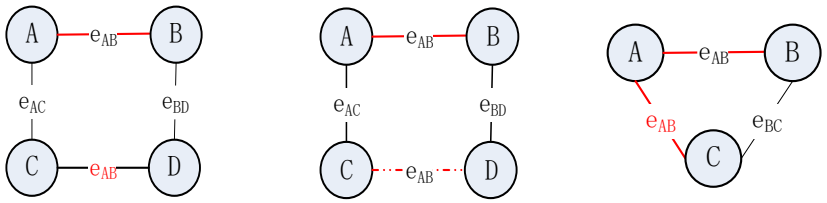


Fig. 4. Sensitive edge perturbation operations

Based on above discussion, our proposed privacy perturbation method for weighted sensitive edges is divided into two parts: (1) calculate the edge betweenness coefficient in the shortest path set, and filter out the sensitive edges according to the threshold, which is implemented by algorithm1. (2) discuss the neighbor structure of each sensitive edge, and then perform the privacy perturbation operations by adding new pseudo edges or modifying the weight of adjacent edges, we develop algorithm 2 to implement it.

Algorithm1. Edge betweenness computing and sensitive edge filtering

Input: Network graph G=(V, E), shortest path set P, and sensitive threshold λ

Output: Sensitive edge set S;

```
1 for each e_i ∈ E do
2  for each p_j ∈ P do //for each path p_j in the path set P
3                  for each s∈ p_j do //for each edge s of the path p_j
4                              if e_i ==s then
5            e_i.BC++; //to get the number of paths through edge e_i
6             break;
7           endif
8                   end for
9   end for
10 end for
11 for each e_i ∈ E do
12                  e_i.BC = e_i.BC /|P|; //Compute the edge betweenness by (1)
13     If e_i.BC ≥ λ then
14       add e_i → S;   //Add sensitive edge e to set S
15 end for
```

In algorithm1, the shortest path set P of the input network G is generated by using the classical Floyd-Warshall method, suppose the scale of the path set P is n, and the average path length in the set is k. Notice that the goal of steps①-⑩ is to count the number of occurrences of each edge in the path set P in the network, the time complexity is $O(|E|\cdot n\cdot k)$. After that, steps⑪-⑮ aim to compute the edge betweenness for each edge and filter out the sensitive edges according to the threshold, the required runtime is $O(|E|)$. Therefore, the overall time complexity of our algorithm is $O(|E|\cdot n\cdot k+|E|)$.

Algorithm2. Weighted sensitive edge perturbation privacy protection

Input: Network graph G=(V, E), Sensitive edge set S

Output: privacy network graph G' after perturbation

```
1 for each e_xy ∈S do
2  if |degree(v_x)|>1 && |degree(v_y)|==1 then // for the case Fig. 2
3                  select vertex v_j with Min{e_xj} // select vertex v_j from the neighbor set of v_x;
4            add a new pseudo edge(v_x, v_j) with the weight e_xj = e_xy;
5  end if
6  if |degree(v_x)|>1 && |degree(v_y)|>1 then
7    for each v_i ∈neighbor(v_x) do
8                  for each v_j ∈neighbor (v_y) do
9          if v_i <> v_j && edge(v_i, v_j)∈ E then  //situation of Fig. 4(a)
10              modify the weight of edge(v_i, v_j)  e_ij = e_xy;
11          if v_i <> v_j && edge(v_i, v_j)∉ E then  //situation of Fig. 4(b)
```

```
12                    add new pseudo edge( v_i , v_j ) and assign e_ij = e_xy ;
13          if v_i <> v_j && e_xy + e_yi < e_xi then  // situation of Fig. 4(c)
14              modify the weight of edge( v_x , v_i )and assign e_xi = e_xy ;
15          neighbor ( v_y )=neighbor ( v_y )-{ v_j };  //remove from neighbor set
16          break;
17      end for
18      neighbor ( v_x )=neighbor ( v_x )-{ v_i };
19    end for
20  end if
21 end for
```

The algorithm 2 has implemented the perturbation operations dealing with various situations of the sensitive edges as shown in **Fig. 2** and **Fig. 4** respectively. Among which, for the results in **Fig. 2**, steps②-⑤ perform the perturbation operations by adding new pseudo edge to the neighbor of one side of the sensitive edge, and its weight is assigned same to the sensitive weight. Next, as to the results in **Fig. 4**, steps⑥-⑳ perturb the neighboring vertices of sensitive edges by adding pseudo edge or modifying edge weights of neighbors at both sides of the sensitive edge. To evaluate the computation cost, assume that the size of sensitive edge set is *m*, and the average degree of sensitive edges is *d*. Obviously, we observe that the running time of algorithm 2 is mainly focus on the perturbation operation of the neighbors of the sensitive edges by step⑥-⑳ in two nested loops, and since that step ⑯ will jump out of the inner loop after every perturbation operation is completed. Thus, the overall time complexity required for the algorithm should be $O(m \cdot d)$.

5. Experiment Evaluations

In this section, we provide extensive experiments to evaluate the effects of our privacy perturbation algorithms. All algorithms are implemented using Python3.5, and the experiments are conducted on computers with intel core i7 processor, 8G memory and Windows 8 operating system. In our experiments, we choose three different scales of social network datasets to perform experimental comparisons. The Les Miserables abbreviated as Lesmis, belongs to a typical small-sized undirected miscellaneous network, which contains co-occurence of characters in Victor Hugo's novel 'Les Miserables' including 254 co-occurence relationships among 77 characters. Netscience dataset is a middle-sized network of co-authorships in the area of network science before 2006, containing 2724 co-authorship for 1589 authors. Geom dataset is a large collaboration network in computational geometry with 7343 vertices and 11898 edges. All the other related parameters are shown in **Table 1**.

Table 1. Three different scale social network attributes

Network dataset	#vertices	#edges	Average degree	Average edge weight	Connection Type	Degree distribution
Les Miserables(Lesmis)	77	254	6.60	28	Undirected	power law
Netscience	1589	2724	3.43	40	Undirected	power law
Geom	7343	11898	3.24	73	Undirected	power law

In order to conveniently visualize the experimental results, we choose the relative small dataset Lesmis to demonstrate our perturbation effects on the weighted sensitive edges. **Fig. 5** shows the privacy protection results for sensitive edges in Lesmis dataset when the betweenness threshold λ =0.5. From this graph visualization, we can observe that there exists about 250 non-sensitive edges depicted in green color, and only 4 sensitive edges in red color need to be considered perturbation. After privacy protection, we add and modify about 40 perturbated weight edges marked with blue color, which all are the adjacent edges of these sensitive edges. As a result, the difficulty of malicious attacks against sensitive edges will roughly increase about 10 times than before perturbation.

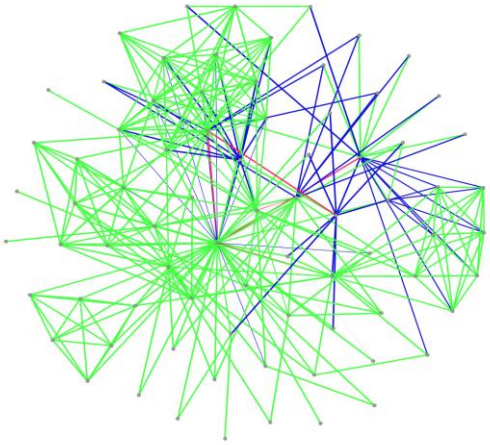


Fig. 5. Perturbation effects on Lesmis dataset when betweenness λ =0.5

Next, we further compare four most popular metrics based on graph characteristics after privacy perturbation on three different datasets. **Fig. 6** shows the changes of average path length after perturbation operations under various thresholds of edge betweenness. The results indicate that the average path length gradually increases with growing edge betweenness in Netscience and Geom datasets, while the growth almost is kept stable in dataset Lesmis. This is because Lesmis belongs to one typical small-scale dataset, and the number of sensitive edges filtered under different betweenness thresholds is so few that sensitive edges are almost kept fixed. Thus, the perturbation operations have little effect on the original network structure.

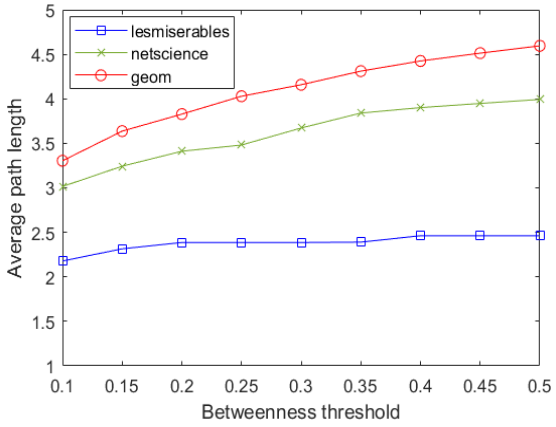


Fig. 6. Changes of average path length after perturbation

Fig. 7 depicts how the perturbation operations influence the average degree of network under various edge betweenness thresholds. Obviously, when the betweenness threshold is larger, the fewer sensitive edges are obtained, and consequently the fewer edges need to perform perturbation operation. As a result, the average degree of network is gradually decreased and nearly approximate to the original unperturbed network. On the whole, when the sensitive edge threshold sits in the interval [0.2, 0.5] in Netscience and Geom datasets, our edge perturbation operations will have less impact to the network structure, and the average degree of perturbed network declines slowly with the slight increasing of sensitive edges. As for a small dataset, for instance Lesmis, the perturbation operations have greater impact on the average degree of network. But when the edge sensitive edge threshold lies in the two intervals, such as [0.2, 0.35] and [0.4, 0.5], the number of sensitive edges does not change much, so the average degree of perturbed network is kept relatively stable.

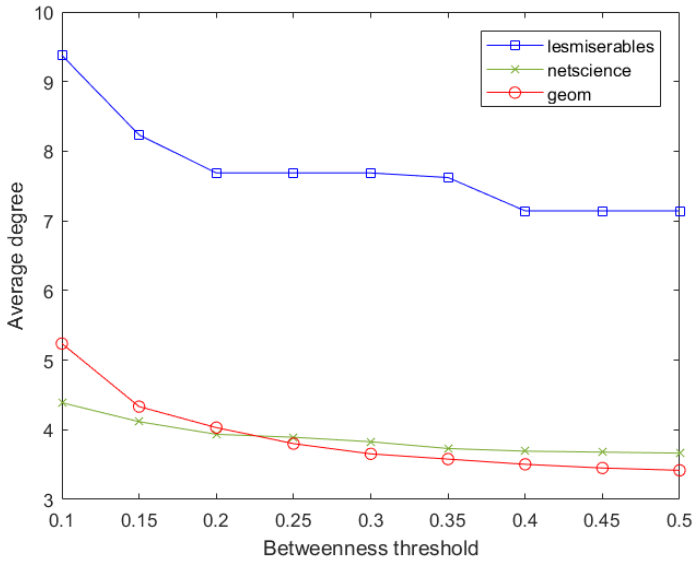


Fig. 7. Impact of average degree of network after edge perturbation

Fig. 8 further shows the results of clustering coefficient changes after network perturbation under various edge betweenness thresholds. Among these datasets, we can observe that the clustering coefficient on dataset Netscience has the most flat changes. The reason is that a large number of perturbation operations belong to modifying the weights of sensitive edges, thus having a minimal impact on the original network structure. While on datasets Lesmis and Geom, there exists gradually increasing number of sensitive edges when the betweenness threshold lies in the interval [0.1, 0.2], consequently the edge perturbation operations make the greater increases of clustering coefficient of reconstructed graph, whereas the overall growth indicates a relatively flat trend. Thus, this shows that our proposed privacy protection for sensitive edges based on perturbation approach could maintain network structure properties relatively stable.

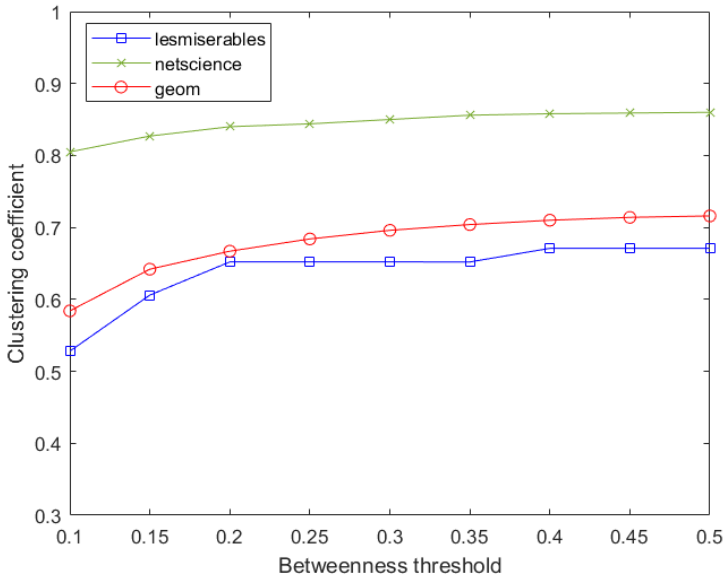


Fig. 8. Impact of clustering coefficient of network after edge perturbation

Fig. 9 reports the number of perturbed edges including adding and modifying operations under various sensitive thresholds. Since Geom is the largest dataset, the total number of the sensitive edges required to be perturbed always records the largest magnitude, and will drop significantly as the edge betweenness threshold increases slowly. While in Netscience and Lesmis datasets, the number of edges needed to be perturbed decreases very gently with the increase of edge betweenness threshold, there are no obvious fluctuations compared with Geom dataset. The main reason is that Netscience and Lesmis belong to medium and small scale datasets respectively, so the number of sensitive edges does not have the same order of magnitude as Geom.

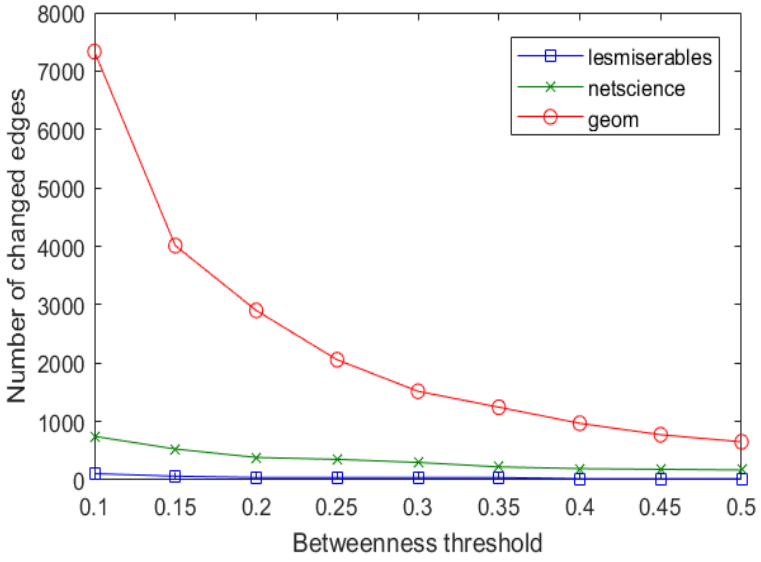


Fig. 9. Count of changed edges in perturbation operations

Besides measuring above properties of network structure, we conduct evaluation of the possibility of inference attacks and the information loss caused by perturbation. **Fig. 10** shows the results of attack success rate under various sensitive thresholds on three datasets. For Lesmis dataset, the success rate of attacks goes up scalariformly when the edge betweeness thresholds are in the two intervals, for instance [0.2, 0.35] and [0.4, 0.5]. The reason is that the number of sensitive edges has minor changes in these two intervals, hence the success rate attacking at sensitive edges is almost kept unchanged after perturbation. While in the cases of Netscience and Geom datasets, the success rate of attacks goes up slightly as the sensitive edge threshold increases, but they both are remained at a very low level. Therefore, the results show that our perturbation based approach on different social network datasets can achieve ideal privacy protection effects for sensitive edges.

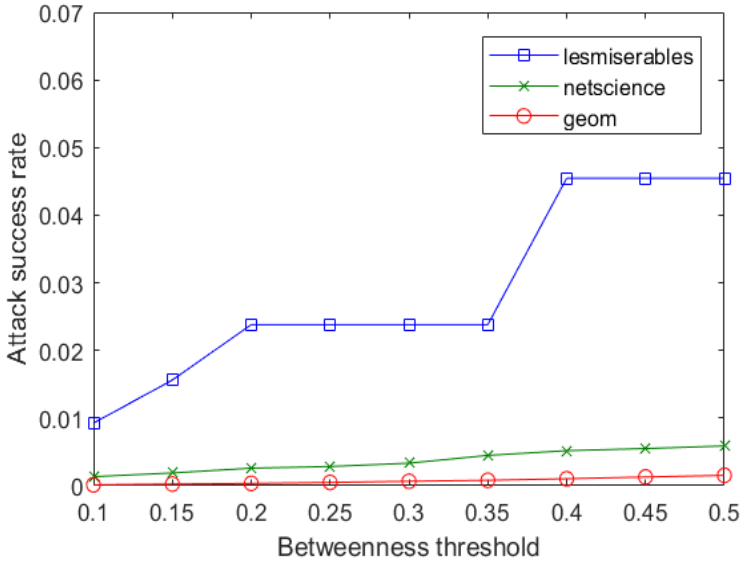


Fig. 10. Attack success rate under various sensitive thresholds

Finally, **Fig. 11** and **Fig. 12** summarize the quantity of information loss and the loss rate before and after network perturbation. As defined in (4), the computation of information loss is not only including the sum of all weights of added edges but also the accumulative weights of modified edges, and here the information loss rate refers to the proportion of information loss quantity to the total weights of original network.

As shown in **Fig. 11**, the information loss of Geom after perturbation always keeps the largest quantity among these three datasets, and its tendency is to decrease sharply with the increase of sensitive edge threshold, whereas in case of Lesmis and Netscience datasets, their information loss maintains a steady downward trends. This is because the size of dataset Geom is much larger than the other two datasets, and besides that Geom also has the largest average edge weight. As mentioned above, similar results are depicted in **Fig. 9**, it is indicated that the number of changed edges by perturbation in dataset Geom is far more than that in Lesmis and Netscience datasets. At the same time, the results in **Fig. 12** are shown that when the sensitive edge threshold is greater than 0.3, the information loss rate in these three datasets changes approximately same to each other. Overall, we can conclude that our privacy protection approach for sensitive edges in these datasets can make the information loss caused by perturbation being controlled within the ideal range.

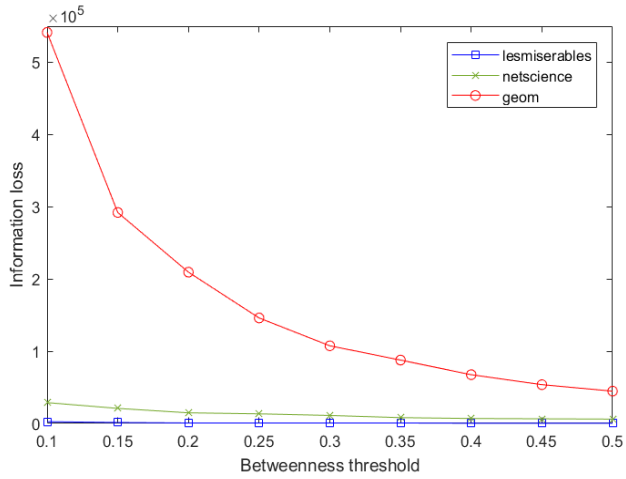


Fig. 11. Information loss of edge weights

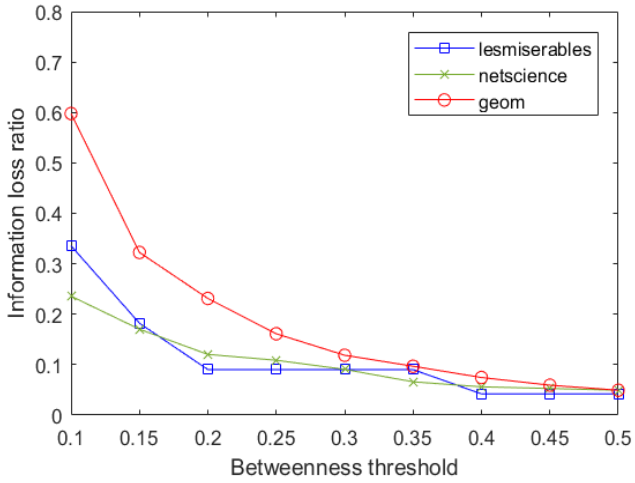


Fig. 12. Ratio of information loss

6. Conclusion

This paper mainly addresses the privacy issues of sensitive edges in weighted social graph. In order to identify the high valuable and important edges for privacy preserving, we give the formal notion of sensitive edges considering as privacy data which are differentiated from the normal weighted edges in the social network graph. Then the privacy preservation based on graph perturbation approach is proposed to protect the privacy of important and sensitive areas in social networks by adding pseudo-edges or modifying the weights of sensitive edges. In addition, we provide the metric of information loss to evaluate the cost of privacy preservation, and conduct extensive experiments on three typical real-world datasets. The experimental results demonstrate that our proposed perturbation method can not only keep relatively stable structural properties of network graph with lower cost, but also effectively provide satisfactory privacy guarantee to defend against malicious attacks to important sensitive edges of social network. As a future work, we plan to investigate superior perturbation techniques to ensure the privacy for more complicated sensitive edges with multi-dimensional properties.

References

[1] X. Li, C. Zhang, T. Jung, J. Qian, and L. Chen, "Graph-based privacy-preserving data publication," in *Proc. of the 35th Annual IEEE International Conference on Computer Communications(INFOCOM)*, pp. 1-9, 2016. Article (CrossRef Link)

[2] A. Zaghian and A. Bagheri, "A combined model of clustering and classification methods for preserving privacy in social networks against inference and neighborhood attacks," *International Journal of Security and its Applications*, vol. 10, no. 1, pp. 95-102, 2016. Article (CrossRef Link)

[3] Y. Lv, T. Ma, M. Tang, J. Cao, Y. Tian, A. Dhelaan, and M. Rodhaan, "An efficient and scalable density based clustering algorithm for datasets with complex structures," *Neurocomputing*, vol. 171, pp. 9-22, 2016. Article (CrossRef Link)

[4] A. Campan and M. Truta, "Data and structural *k*-anonymity in social networks," in *Proc. of the 2nd ACM SIGKDD International Workshop on Privacy, Security, and Trust in KDD(PinKDD'08)*, vol. 5456, pp. 33-54, 2008. Article (CrossRef Link)

[5] F. Yu, M. Chen, B. Yu, W. Li, L. Ma, and H. Gao, "Privacy preservation based on clustering perturbation algorithm for social network," *Multimedia Tools and Applications*, vol. 77, no. 9, pp. 11241-11258, 2018. Article (CrossRef Link)

[6] Z. Wang, X. Pang, Y. Chen, H. Shao, Q. Wang, L. Wu, H. Chen, and H. Qi, "Privacy preserving crowd-sourced statistical data publishing with an untrusted server," *IEEE Transactions on Mobile Computing*, vol. 18, no. 6, pp. 1356-1367, 2019. Article (CrossRef Link)

[7] B. Francesco, G. Aristides, and T. Tamir, "Identity obfuscation in graphs through the information theoretic lens," *Information Sciences*, vol. 275, pp. 232-256, 2014. Article (CrossRef Link)

[8] C. Tai, P. Yu, D. Yang, and M. Chen, "Privacy-preserving social network publication against friendship attacks," in *Proc. of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining(KDD'11)*, pp. 1262-1270, 2011. Article (CrossRef Link)

[9] X. Ying and X. Wu, "Randomizing social networks: a spectrum preserving approach," *in Proc. of the SIAM International Conference on Data Mining (SDM'08)*, pp. 739-750, 2008. Article (CrossRef Link)

[10] X. Ying and X. Wu, "On link privacy in randomizing social networks," *Knowledge and Information Systems*, vol. 28, no. 3, pp. 645-663, 2011. Article (CrossRef Link)

[11] E. Zheleva and L. Getoor, "Preserving the privacy of sensitive relationships in graph data," in *Proc. of the 1st ACM SIGKDD Workshop on Privacy, Security, and Trust in KDD (PinKDD'07)*, vol. 4890, pp.153-171, 2007. Article (CrossRef Link)

[12] M. Hay, K. Liu, G. Miklau, J. Pei, and E. Terzi, "Privacy-aware data management in information networks," in *Proc. of International Conference on Management of Data(SIGMOD)*, pp. 1201-1204, 2011. Article (CrossRef Link)

[13] C. Jordi, H. Jordi, and V. Torra, "A survey of graph-modification techniques for privacy-preserving on networks," *Artificial Intelligence Review*, vol. 47, no. 3, pp. 341-366, 2017. Article (CrossRef Link)

[14] M. Hay, G. Miklau, D. Jensen, D. Towsley, and C. Li, "Resisting structural re-identification in anonymized social networks," *The VLDB Journal*, vol. 19, no. 6, pp. 797-823, 2010. Article (CrossRef Link)

[15] E. Maria, M. Manolis, G. Stefanos, M. Lilian, T. Hannu, and M. Pirjo, "Privacy preservation by k-anonymization of weighted social networks," in *Proc. of the International Conference on Advances in Social Networks Analysis and Mining(ASONAM)*, pp. 423-428, 2012. Article (CrossRef Link)

[16] S. Bhagat, G. Cormode, B. Krishnamurthy, and D. Srivastava, "Class-based graph anonymization for social network data," *the VLDB Endowment (PVLDB)*, vol. 2, no. 1, pp.766-777, 2009. Article (CrossRef Link)

[17] L. Zou, L. Chen, and M. Tamer, "K-automorphism: A general framework for privacy preserving network publication," *the VLDB Endowment (PVLDB)*, vol. 2, no. 1, pp. 946-957, 2009. Article (CrossRef Link)

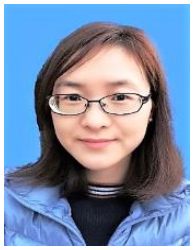
[18] J. Cheng, A. Fu, and J. Liu, "K-Isomorphism: Privacy preserving network publication against structural attacks," in *Proc. of the ACM SIGMOD International Conference on Management of Data*, pp. 459-470, 2010. Article (CrossRef Link)
[19] X. Liu and X. Yang, "A generalization based approach for anonymizing weighted social network graphs," in *Proc. of the 12th International Conference on Web-age Information Management(WAIM'11)*, pp. 118-130, 2011. Article (CrossRef Link)
[20] M. Yuan and L. Chen, "Node protection in weighted social networks," in *Proc. of the 16th International Conference on Database Systems for Advanced Applications(DASFAA 2011)*, pp. 123-137, 2011. Article (CrossRef Link)
[21] Y. Li and H. Shen, "Anonymizing graphs against weight-based attacks," in *Proc. of IEEE International Conference on Data Mining Workshops (ICDMW 2010)*, pp. 491-498, 2010. Article (CrossRef Link)
[22] Q. Xiao, R. Chen, and K. Tan, "Differentially private network data release via structural inference," in *Proc. of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 911-920, 2014. Article (CrossRef Link)
[23] L. Liu, J. Wang, J. Liu, and J. Zhang, "Privacy preservation in social networks with sensitive edge weights," in *Proc. of the SIAM International Conference on Data Mining (SDM'09)*, pp. 954-965, 2009. Article (CrossRef Link)
[24] S. Das, O. Egecioglu, and A. Abbadi, "Anonymizing weighted social network graphs," in *Proc. of IEEE 26th International Conference on Data Engineering (ICDE 2010)*, pp. 904-907, 2010. Article (CrossRef Link)
[25] M. Yuan, L. Chen, P. Yu, and T. Yu, "Protecting sensitive labels in social network data anonymization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 3, pp. 633-647, 2013. Article (CrossRef Link)
[26] X. Liu and X. Yang, "Protecting sensitive relationships against inference attacks in social networks," in *Proc. of the 17th International Conference on Database Systems for Advanced Applications(DASFAA2012)*, pp. 335-350, 2012. Article (CrossRef Link)

Weihua Gong received the Ph.D. degree in computer software and theory from Huazhong University of Science and Technology, Wuhan, China, in 2006. He is currently a associate professor in School of Computer Science and Technology, Zhejiang University of Technology, China. His research area includes data mining, social networks, and machine learning.

Rong Jin received the M.S. degree from Hubei University, Wuhan, China, in 2005. She is currently an Lecturer in School of Informatics and Electronics, Zhejiang Sci-Tech University, Hangzhou, China. Her main research interests include social networks, recommender system and data mining.

Yanjun Li received the B.S. and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 2004 and 2009, respectively, and another Ph.D. degree from Nancy University, Villers-les-Nancy, France, in 2010. She is currently a Professor in School of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China. Her research area includes privacy protection and internet of things. She has published more than 70 referred technical papers in proceedings and journals.

Lianghuai Yang received the Ph.D. degree from Peking University, Beijing, China, in 2001. He is currently a Professor in School of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China. His research area includes database system, big data computing and data mining.



Jianping Mei received the Ph.D. degree from Nanyang Technological University, Singapore, in 2012. She is currently an Associate Professor in School of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China. Her research area includes data mining and machine learning.