

Privacy Preserving in Social Networks Against Sensitive Edge Disclosure

Lian Liu ^{#1}, Jie Wang ^{#2}, Jinze Liu ^{*3}, Jun Zhang ^{#4}

[#]Laboratory for High Performance Scientific Computing and Computer Simulation,
Department of Computer Science, University of Kentucky, Lexington, KY 40506-0046, USA

¹lliuc@csr.uky.edu ²jwanga@csr.uky.edu ⁴jzhang@cs.uky.edu

^{*}Department of Computer Science, College of Engineering,
University of Kentucky, 237 Hardyman, Lexington, KY, 40506-0046, USA

³liuj@cs.uky.edu

Abstract—With the development of emerging social networks, such as Facebook and MySpace, security and privacy threats arising from social network analysis bring a risk of disclosure of confidential knowledge when the social network data is shared or made public. In addition to the current social network anonymity de-identification techniques, we study a situation, such as in business transaction networks or intelligence communities (terrorist networks), in which the network edges (transaction cost or terrorist contact frequency) as well as the corresponding weights are considered to be private. We consider preserving weights (data privacy) of some edges, while trying to preserve close shortest path lengths and exactly the same shortest paths (data utility) of some pairs of nodes. We develop two privacy-preserving strategies for this application. The first strategy is based on a Gaussian randomization multiplication, and the second one is a greedy perturbation algorithm which is based on the graph theory. Especially, the second strategy can not only keep a close shortest path length and exactly the same shortest path for certain selected paths, but also maximize the weight privacy preservation, demonstrated by our mathematical analysis and experiments.

I. INTRODUCTION

A social network is a special graph structure made of entities and connections between these entities. These entities, or nodes, are abstract representations of either individuals or organizations that are connected by one or more attributes. The connections, or edges, denote relationships or interactions between these nodes. Connections can be used to represent financial exchange, friend relationships, conflict likelihood, web links, sexual relationships, disease transmission (epidemiology), etc. Although studying social networks has wide applications and attracted more and more attentions in recent years, we still face the challenge of achieving a reasonable tradeoff between securing the confidential information associated with the social networks and maximizing the benefits from the social network analysis. These threats against privacy of the social networks promote us to develop social network privacy oriented -preserving techniques.

Recently, social networks have been an undoubted hotspot in data mining communities since advances in computer and network have made it easy to gather and collect data based on different persons and organizations, ranging from epidemiol-

ogists [14], [17], sociologists [18], zoologists [5], [8], [19], to intelligence communities (terrorism networks) [3], and much more. Much progress has been made in degree distribution (the degree of a node tells how many edges connect this node to other ones), network topology (isomorphism), growth models (network temporal attraction to new members), small-world effect (the average shortest path length for social networks is empirically small), community identification (functional group transformation), etc. With the continuous development of social network applications, the need to protect confidential, sensitive, and security information from being disclosed should be considered.

Some researchers have already studied problems in privacy preserving social networks. Most of them focus on the de-identification process to protect the privacy about explicit individuals while preserving the patterns between small communities [11], [25], [26]. In their work, social networks to be preserved are given as unweighted graphs in which links are just a relationship without other meanings.

In fact, edge weights, reflecting affinity between two nodes in many cases, relate the expenses or frequency between two persons or similarity between two organizations. The edge weights in the network are more realistically assigned on a practical scale. For example, in a protein interaction network [12], the weight, called the interaction strength, represents the probability that two proteins have a synergistic interaction.

A more realistic example is the business network, in which the edge probably denotes the transaction expenses according to some measures (such as per month, per person or per transaction) between the two linked companies [24]. Due to the competition between companies, most managers may not be willing to disclose the true transaction expenses to their adversaries (otherwise, their adversaries probably reduce the quotation below the price in a secret bidding competition). Hence, they would like to perturb their transaction expenses (edge weights) before publication of related social networks. But, at the same time, some global and local utilities, such as the shortest paths and the corresponding lengths, of social networks are probably desired to be maintained for future analysis. In a business example, for example, Company A

wants to purchase some products or services, in the future, from Company D which is not possible to directly access each other due to some trade barriers. Company A needs choose some trade intermediate agents who have the most competitive path (the shortest path of price) between themselves and Company D (maybe these agents need other agents to connect Company D). If the edges of the business social networks are perturbed but the shortest paths (and the corresponding lengths) are well preserved, Company A may be able to make an intelligent decision based on this privacy-preserved social networks.

In addition to the above example of the shortest path utility, the shortest path is important to be preserved in a social network for the following reasons. 1), Previous work is mostly on the unweighted graph. Their work is mostly focused on de-identification of nodes or edges. 2), The weighted graph is quite popular. One of the things people care about in this type of graphs is the shortest path between every pair of nodes. The shortest path is a major data utility which has a wide application such as physical location search in GIS, min-delay path problem in telecommunications midset, and optimal Analog circuits in VLSI (very large scale integration). 3), In essence, a weighted graph is a generalization of the unweighted graph. Our algorithms might be generalized and extended to unweighted graph cases.

So, in this paper, we consider preserving weights (data privacy) of some edges, while trying to preserve close shortest path lengths and exactly the same shortest paths (data utility) of some pairs of nodes without adding or deleting any edge and node.

The remaining parts of this paper are arranged as follows. A brief introduction to the related work and some popular data perturbation techniques are in Section II. Two edge privacy-preserving strategies and our theoretical analyses are presented in Section III. Experimental results are listed and discussed in Section IV. Finally a brief conclusion is given in Section V.

II. RELATED WORK

In privacy-preserving data mining, various techniques have been developed to maintain the data utility without disclosing the original data and guarantee that the data mining analysis results are as close to those based on the original data as possible. Generally, among various privacy-preserving data mining and analysis techniques, we mention two main categories. Methods in the first category modify data mining algorithms so that they allow data mining operations on distributed datasets without knowing the exact values of the data or without direct access to the original datasets. Methods in the other category perturb the values of the datasets to protect privacy of the data values. These methods are designed to perturb the whole dataset or the confidential parts of the dataset using matrix decomposition or signal processing techniques [2], [13], [15], [22], [23] and randomization addition [6], [16].

In social networks, the data is not meaningfully represented by a tabular or matrix. Hence, most people do not use traditional matrix-based algorithm to preserve privacy. They

emphasize the protection of social entity's identification via de-identification techniques [21]. For example, Hay et al. [11] and Zhou et al. [26] presented a framework to add and delete some unweighted edges in social networks to prevent attackers from accurately re-identifying the nodes based on background information about the neighborhood. Read et al. [17] and Rogers [18] theoretically defined a family of attacks based on random graph theory and link mining prospect. They first added some distinguishable nodes into the social network before it is collected and published, and after that they used the known added nodes to differentiate the original graph patterns. Zheleva et al. [25] proposed a model in which nodes are not labeled but edges are labeled which are sensitive and should be hidden. They hid and removed some edges based on edge clustering techniques.

The above methods all focus on preserving either node or edge privacy. In this paper, we emphasize edge weight privacy. Data owners may not want to release the exact weight of each edge, but would like to keep the shortest paths of a set of nodes and the lengths of the corresponding shortest paths as unperturbed as possible, for the data analysis purpose.

III. EDGE WEIGHT PERTURBATION

There exist a variety of social networks. Some of them are dynamic in which a social network will develop continuously and its structure may become very large and unpredictable. The other ones are static which may not change dramatically in a short period time.

Due to the difficulty of collecting global information about the social networks in the first category, we develop a Gaussian randomization multiplication technique which does not need any network information in advance. On the other hand, a static social network is the one that we may easily obtain useful structural information such as the existing shortest paths and the corresponding path lengths in advance. With this information, we can develop a useful edge weight perturbation strategy based on a greedy perturbation algorithm to achieve our goal.

We give some notations about our strategies at first, and then introduce our two strategies, Gaussian randomization multiplication and greedy perturbation algorithm, which serve different purposes. The Gaussian method mainly focuses on preserving the length of perturbed shortest paths within some boundaries of the original ones but not guaranteeing the same shortest path after perturbation in a dynamic environment. The advantages of the greedy perturbation algorithm over the Gaussian algorithm are that it can keep the same shortest path after perturbation as the original shortest path before the perturbation, in addition to keep the shortest path length close to that of the original one.

A. Notation

A social network in this paper is defined as an undirected and weighted graph $G=\{V, E, W\}$. Figure 1 is a simple social network. The nodes of the graph, V , may denote meaningful

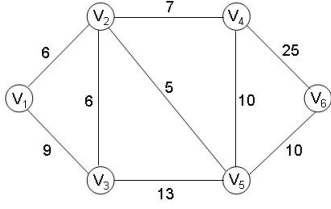


Fig. 1. A simple social network G

entities from the real world such as individuals, organs, organizations, communities, and so on (in Figure 1, $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$). E is the set of all undirected but weighted edges. The edge weight between node i and node j is $w_{i,j}$ (the value beside an edge is the weight in Figure 1). All $w_{i,j}$ compose the set W . The cardinalities of V and E , $\|V\|$ and $\|E\|$, are the number of nodes and edges in this social network, respectively, (in the example, $\|V\|=6$ and $\|E\|=9$). We assume that $n=\|V\|$, $m=\|E\|$. Since the graph G is undirected, $w_{i,j}$ is equal to $w_{j,i}$. So the adjacency weight matrix of G is symmetric. Although the following perturbation strategies are all based on the undirected graph and symmetric adjacency weight matrix, they can be easily modified with respect to directed graphs and the corresponding nonsymmetric adjacency weight matrices.

Let $w_{i,j}^*$ be the perturbed weight of the edge between node i and node j , $d_{i,j}$ and $d_{i,j}^*$ be the shortest path lengths between node i and node j before and after a perturbation strategy, respectively, $p_{i,j}$ and $p_{i,j}^*$ be the shortest paths between node i and node j before and after a perturbation strategy.

B. Gaussian Randomization Multiplication

In this section, we describe some preliminaries and the intuition behind our edge weight perturbation strategy in a social network represented as an undirected but weighted graph without loops and multiedges.

Proposition 1. *There does not exist a perturbation schema such that every edge weight is perturbed but the length of the shortest paths between every pair of nodes are preserved.*

Proof: By contradiction.

Let $e_{i,k_1}, e_{k_1,k_2}, \dots, e_{k_{h-1},k_h}, e_{k_h,j}$ be the shortest path between node i and node j (their corresponding weights are $w_{i,k_1}, w_{k_1,k_2}, \dots, w_{k_{h-1},k_h}, w_{k_h,j}$). We assume that there is a perfect perturbation strategy which perturbs each edge weight but preserves every node pair's shortest path length. Obviously, the path $e_{i,k_1}^*, e_{k_1,k_2}^*, \dots, e_{k_{h-1},k_h}^*$ is the shortest path between nodes i and k_h which can be easily proved by contradiction (subpaths of the shortest paths are the shortest paths, see pp. 519 of [4]), and $d_{i,k_h} = d_{i,k_h}^*$. It follows that

$$\begin{aligned} d_{i,j}^* &= d_{i,k_h}^* + w_{k_h,j}^* \\ &= d_{i,k_h} + w_{k_h,j}^* \\ &\neq d_{i,k_h} + w_{k_h,j} \\ &= d_{i,j} \end{aligned}$$

Hence, our assumption at the beginning of the proof is incorrect. Namely, there does not exist such a perfect perturbation schema. ■

Gaussian randomization multiplication strategy. We assume that W is an $n \times n$ matrix whose entries are either weights if two nodes have a link or ∞ otherwise. W is called the adjacency weight matrix of the graph G . W^* is the perturbed adjacency weight matrix with the same dimension after our schema. $N(0, \sigma^2)$ stands for an $n \times n$ symmetric Gaussian noise matrix with the mean 0 and the standard deviation σ . We define the perturbed weight of each edge as

$$w_{i,j}^* = w_{i,j}(1 - x_{i,j}), \quad i, j = 1, \dots, n.$$

Here $x_{i,j}$ is a randomly generated number from the Gaussian distribution $N(0, \sigma^2)$. The Gaussian-perturbed version of the graph G in Figure 1 is shown in Figure 2. Here, the symmetric Gaussian noise matrix is generated from $N(0, 0.15^2)$ ($\sigma=0.15$).

The reasons why we chose the Gaussian randomization multiplication strategy are as follows. 1). It is very easy to implement in practice. 2). Due to the dynamic evolution nature of social networks, it is very hard or costly to collect all global information in advance in a huge and dynamic social network. In particular, in an evolutionary environment, some nodes or edges in the future will emerge and be added to the current network, in which the collection of the current state will probably be totally changed after these insertions. So it is impossible or not very useful to collect comprehensive global information at a given time for later analysis.

We can reconstruct the perturbed graph $G^* = \{V^*, E^*, W^*\}$. It is clear that the above Gaussian randomization multiplication strategy does not change the structure of the original graph. Namely, $V = V^*$, $E = E^*$. The only difference between G and G^* is the weights.

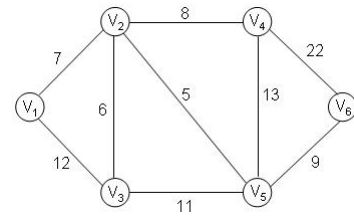


Fig. 2. The perturbed social network G^* of G in Figure 1

Figures 1 and 2 are obviously simple static social networks rather than a dynamic one, and we only use them as an illustration of our ideas and algorithms. In Figure 2, all values of V^* and E^* are clearly the same as those of V and E in Figure 1. The only difference between G^* and G in our figures is the numbers corresponding to the weights.

For most paths in the network, using Gaussian randomization multiplication will keep a perturbed shortest path length close to the original one within a small range, 2σ , as shown in Theorem 2.

Theorem 2. *In the Gaussian randomization multiplication strategy,*

$$\Pr\left(\frac{|d_{i,j}^* - d_{i,j}|}{d_{i,j}} \leq n\sigma\right) \geq \operatorname{erf}\left(\frac{n}{\sqrt{2}}\right), \text{ for every different } i, j$$

Proof: $\Pr\left(\frac{|d_{i,j}^* - d_{i,j}|}{d_{i,j}} \leq n\sigma\right)$ is the probability function of $\frac{|d_{i,j}^* - d_{i,j}|}{d_{i,j}}$ being smaller than $n\sigma$. $\operatorname{erf}(\Delta)$ is the Gaussian error function. $d_{i,j} = w_{i,k_1} + w_{k_1,k_2} + \dots + w_{k_h,j}$, and $x_{i,j}$ is a randomly generated number from the Gaussian distribution $N(0, \sigma^2)$. Let $u = \max(|x_{i,j}|)$. According to our perturbation strategy, we have

$$\begin{aligned} w_{i,k_1}^* &= w_{i,k_1} * (1 - x_{i,k_1}), \\ w_{k_1,k_2}^* &= w_{k_1,k_2} * (1 - x_{k_1,k_2}), \\ &\dots \\ w_{k_h,j}^* &= w_{k_h,j} * (1 - x_{k_h,j}). \end{aligned}$$

Sum up the above equations,

$$\begin{aligned} d_{i,j}^* &\geq d_{i,j} * (1 - u), \\ d_{i,j}^* - d_{i,j} &\geq -d_{i,j} * u, \\ d_{i,j} - d_{i,j}^* &\leq d_{i,j} * u, \\ \frac{|d_{i,j}^* - d_{i,j}|}{d_{i,j}} &\leq u. \end{aligned} \quad (1)$$

Take the probability function on both side of Inequality (1), we obtain

$$\Pr\left(\frac{|d_{i,j}^* - d_{i,j}|}{d_{i,j}} \leq n\sigma\right) \leq \Pr(u \leq n\sigma). \quad (2)$$

According to [20], in a Gaussian distribution (u is the maximum value of the absolute numbers generated from a Gaussian distribution), $\Pr(u \leq n\sigma) \leq \operatorname{erf}\left(\frac{n}{\sqrt{2}}\right)$. So, Inequality (2) extends to:

$$\begin{aligned} \Pr\left(\frac{|d_{i,j}^* - d_{i,j}|}{d_{i,j}} \leq n\sigma\right) &\leq \Pr(u \leq n\sigma) \\ &\leq \operatorname{erf}\left(\frac{n}{\sqrt{2}}\right). \end{aligned} \quad (3)$$

We note that the path in question is not required to be the shortest length path, and it could be any path between the two nodes.

From [20], we can easily figure out that $\operatorname{erf}\left(\frac{1}{\sqrt{2}}\right)$, $\operatorname{erf}\left(\frac{2}{\sqrt{2}}\right)$ and $\operatorname{erf}\left(\frac{3}{\sqrt{2}}\right)$ are approximately equal to 0.68, 0.95 and 0.997, respectively. In other words, if we carefully choose the parameter σ , based on the above theorem, we can preserve the weight summations of each path, including the shortest path, as close as possible to those of the original social network while protect the exact edge weights of the original networks from disclosure.

Comparing Figure 1 to Figure 2, we can see that all perturbed shortest path lengths between every node pair except for $d_{1,3}^*$ are in the corresponding range $[d_{i,j}(1-2\sigma), d_{i,j}(1+2\sigma)]$ (here, $\sigma=0.15$). $d_{1,3}$ is 9 and $d_{1,3}^*$ is 12 and the difference is

0.33 which is more than 2σ . In other words, in the totally 15 shortest paths (due to the symmetry, $p_{i,j}$ and $p_{j,i}$ are counted only once), the lengths of 14 perturbed shortest paths are in the range $[d_{i,j}(1-2\sigma), d_{i,j}(1+2\sigma)]$ with the length of just one perturbed shortest path, $p_{1,3}^*$, being outside the range. The ratio of perturbed shortest path lengths falling within the range is $14/15=93\%$ which is consistent with our mathematical analysis in Theorem 2.

Corollary 3. *If the ratio of difference between the shortest path length and the second shortest path length to the shortest path length is greater than 2σ , the shortest path is highly possible to be preserved after our Gaussian randomization multiplication strategy. Here, σ is the parameter of the Gaussian noise matrix $N(0, \sigma^2)$.*

According to the Corollary 3, in the case of a good choice of σ , for example, $\sigma \in [0.1, 0.2]$, we could preserve not only a very accurate shortest path length between certain pairs, but also exactly the same shortest path after our perturbation strategy.

Comparing Figure 1 to Figure 2 again, all perturbed shortest paths, except $p_{3,5}^*$, $p_{4,5}^*$ and $p_{4,6}^*$, are identical with the original ones. In this example, all the three shortest paths have two different paths of equal length, ($p_{3,5}^*=(3 \rightarrow 5)$ or $(3 \rightarrow 2 \rightarrow 5)$, $p_{4,5}^*=(4 \rightarrow 5)$ or $(4 \rightarrow 2 \rightarrow 5)$, $p_{4,6}^*=(4 \rightarrow 6)$ or $(4 \rightarrow 5 \rightarrow 6)$), the second of these is different from the corresponding original ones. Therefore we consider their perturbed shortest paths are changed even one of their perturbed shortest paths is the same as that of the original one.

The original shortest path length between v_3 and v_5 in Figure 1 is 11 ($3 \rightarrow 2 \rightarrow 5$) and the original second shortest path length is 13 ($3 \rightarrow 5$). Its ratio of difference between the shortest path length and the original second shortest path length to the shortest path length is $(13-11)/11=0.18$ which is not greater than 2σ , so the perturbed shortest path may be changed after the Gaussian strategy (actually, in our example, $p_{3,5}^*$ has two different paths which are not considered to be the exactly preserved in comparison to the original $p_{3,5}$ according to our above statement). By contrast, The original shortest path length between v_1 and v_6 in Figure 1 is 21 ($1 \rightarrow 2 \rightarrow 5 \rightarrow 6$) and the original second shortest path length is 30 ($1 \rightarrow 3 \rightarrow 2 \rightarrow 5 \rightarrow 6$). So the perturbed shortest path, $p_{1,6}^*$, is exactly preserved since its ratio of difference between the shortest path length and the second shortest path length to the shortest path length is $(30-21)/21=0.43$ which is greater than 2σ .

But the Gaussian randomization multiplication strategy cannot guarantee the same shortest path preservation after perturbation, if the difference between the shortest path length and the second shortest path length is very small. Therefore, we give another strategy to ensure that, for certain selected shortest paths, the perturbation strategy preserves exactly the same shortest paths in any case in a static social network in the next section.

C. Greedy Perturbation Algorithm

In a static social network, we may easily collect some necessary information about this social network for our analysis and privacy-preserving purpose.

Before applying our perturbation strategy, we should assume that not all shortest paths of node pairs in a social network are considered to be significant (in the real world, it is not reasonable that all information is considered as confidential). In other words, we want to keep certain shortest paths (the starting and ending nodes, (i,j) , in these shortest paths compose a node pair set H , see below) and the corresponding lengths as close to the original ones as possible, while ignore possible changes to other paths. Let H be the set of targeted pairs whose shortest paths and the corresponding path lengths should be preserved as close as possible. For example, in the graph $G=\{V, E, W\}$ in Figure 1, let H be $\{(1,6), (4,6), (3,6)\}$. In a real social network, some shortest paths are just one-edge length paths (e.g., $p_{1,3}=e_{1,3}$), but we assume that these shortest paths are not included in H . In this case, our greedy perturbation algorithm aims to keep exact shortest paths and the corresponding close path lengths between v_1 and v_6 , v_4 and v_6 , v_3 and v_6 , respectively.

Then, in a social network $G=\{V, E, W\}$ ($\|V\|=n$), we generate a shortest path matrix P and the corresponding length $n * n$ matrix D . In the matrix P , each entry p_{s_1, s_2} is a linked list representing the shortest path between v_{s_1} and v_{s_2} . For example, $p_{1,6}=(1 \rightarrow 2 \rightarrow 5 \rightarrow 6)$, it shows that the shortest path $p_{1,6}$ successively passes through v_1, v_2, v_5 and v_6 . In the matrix D , each d_{s_1, s_2} is the length of the shortest path connecting v_{s_1} and v_{s_2} . In the following contents, all node pairs (s_1, s_2) of p_{s_1, s_2} and d_{s_1, s_2} are in the set H unless otherwise stated explicitly.

So, our goal is to generate a perturbed graph $G^*=\{V^*, E^*, W^*\}$ which satisfies the following conditions:

- $V^* = V$ and $E^* = E$,
- maximize $\sum_{i,j} |w_{i,j} - w_{i,j}^*|$,
- $p_{s_1, s_2}^* = p_{s_1, s_2}$, for every pair (s_1, s_2) in H ,
- $d_{s_1, s_2}^* \approx d_{s_1, s_2}$, for every pair (s_1, s_2) in H .

Based on the combination of the above conditions and the collected information, like the matrices P and D , we divide all edges in G into three different categories as in Figure 3.

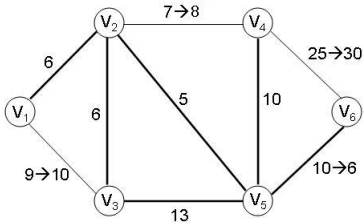


Fig. 3. Three different categories of edges

Definition 4. An edge $e_{i,j}$ is a non-betweenness edge, if $e_{i,j} \notin p_{s_1, s_2}$ for every $(s_1, s_2) \in H$. In other words, no shortest path in P passes through the edge $e_{i,j}$.

In Figure 3, all thin edges such as edges $e_{1,3}, e_{2,4}, e_{4,6}$ and $e_{3,5}$ are non-betweenness edges, because the shortest length paths of all three targeted pairs in $H=\{(1,6), (4,6), (3,6)\}$ do not pass through these edges. In practice, empirically, the non-betweenness edges are the majority of edges in a social network.

Definition 5. We call an edge $e_{i,j}$ an all-betweenness edge, if $e_{i,j} \in p_{s_1, s_2}$ for every $(s_1, s_2) \in H$, (i.e., all shortest paths in H pass through the edge $e_{i,j}$).

In Figure 3, the dashed edge $e_{5,6}$ is the all-betweenness edge since the shortest length paths $p_{1,6}, p_{4,6}$ and $p_{3,6}$ all go through the edge $e_{5,6}$. Based on the common sense, the all-betweenness edges are very rare in a real social network.

Definition 6. An edge $e_{i,j}$ is a partial-betweenness edge, if $e_{i,j} \in p_{s_1, s_2}$ but $e_{i,j} \notin p_{s_3, s_4}$, for some $(s_1, s_2) \in H$ and $(s_3, s_4) \in H$. In this case, only a part of the shortest paths pass through this edge while this edge does not appear in other part of the shortest paths.

The bold-faced edges in Figure 3 are the partial-betweenness edges. For example, $e_{2,5}$ is a partial-betweenness edge since the shortest paths $p_{1,6}$ and $p_{3,6}$ pass through the edge $e_{2,5}$, but $p_{4,6}$ does not go through it.

We perturb each edge in the graph by three different schemes based on these three different categories.

Proposition 7. For a non-betweenness edge $e_{i,j}$, if we increase its weight by any positive value t (the new perturbed weight is $w_{i,j}^* = w_{i,j} + t$), all d_{s_1, s_2} and p_{s_1, s_2} in H will not be changed, ($d_{s_1, s_2}^* = d_{s_1, s_2}$ and $p_{s_1, s_2}^* = p_{s_1, s_2}$).

Proposition 8. For an all-betweenness edge $e_{i,j}$, if we decrease its weight to any positive value (i.e., $w_{i,j}^* = w_{i,j} - t$ and $w_{i,j}^* > 0$), all p_{s_1, s_2} in H will not be affected, but d_{s_1, s_2} will be decreased. Actually, $p_{s_1, s_2}^* = p_{s_1, s_2}$ and $d_{s_1, s_2}^* = d_{s_1, s_2} - t$.

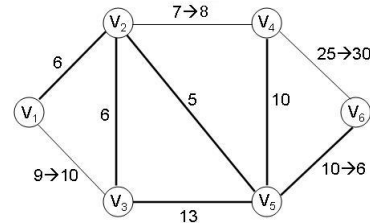


Fig. 4. Perturbation on the non-betweenness and all-betweenness edges

As in the social network shown in Figure 1, we perturb the non-betweenness and all-betweenness edges as in Figure 4. We increase the weights of the non-betweenness edges $e_{1,3}, e_{2,4}$ and $e_{4,6}$, and decrease the weight of the all-betweenness edge $e_{5,6}$.

In a social network, partial-betweenness edges are prevalent which are our major perturbation targets. For the partial-betweenness edges, we have two perturbation schemes.

Proposition 9. For a partial-betweenness edge $e_{i,j}$, we increase its weight by t (the new perturbed weight is $w_{i,j}^* = w_{i,j} + t$) and t satisfies the following condition:

$$t < d_{s_1,s_2}^- - d_{s_1,s_2}, \text{ for all } p_{s_1,s_2} \text{ such that } e_{i,j} \in p_{s_1,s_2}, \quad (4)$$

where d_{s_1,s_2}^- is a conditional shortest path length between node s_1 and node s_2 in a graph $G^- = \{V, E - \{(i,j), (j,i)\}, W - \{w_{i,j}, w_{j,i}\}\}$. G^- is a graph in which we delete the edges $e_{i,j}$ and $e_{j,i}$ and the corresponding weights from G . Of course, for each node pair (s_1, s_2) , $d_{s_1,s_2} \leq d_{s_1,s_2}^-$.

If t satisfies this condition, all p_{s_1,s_2}^* are not changed and d_{s_1,s_2}^* (the edge (i,j) is in p_{s_1,s_2}^*) will become larger, ($p_{s_1,s_2}^* = p_{s_1,s_2}$ and $d_{s_1,s_2}^* = d_{s_1,s_2} + t$).

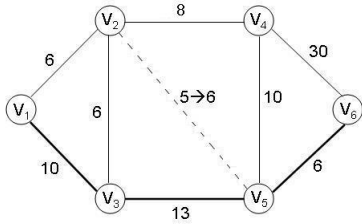


Fig. 5. Increasing the weights of the partial-betweenness edge $e_{2,5}$

An example of increasing the weight of a partial-betweenness edge is $e_{2,5}$ in Figure 5. The two targeted pairs in H , $p_{1,6}$ and $p_{3,6}$, pass through the edge $e_{2,5}$, but the shortest length path $p_{4,6}$ does not go through it. Increasing $w_{2,5}$ will probably affect the shortest paths $p_{1,6}$ and $p_{3,6}$, but has nothing to do with $p_{4,6}$. Hence, there are totally two constraints to increase $w_{2,5}$ to $w_{2,5}^* = w_{2,5} + t$ as follows:

$$\begin{cases} t < d_{1,6}^- - d_{1,6}, \\ t < d_{3,6}^- - d_{3,6}, \end{cases}$$

where $d_{1,6}$ is 17 ($p_{1,6} = (1 \rightarrow 2 \rightarrow 5 \rightarrow 6)$), $d_{1,6}^-$ is 29 ($p_{1,6}^- = (1 \rightarrow 3 \rightarrow 5 \rightarrow 6)$), $d_{3,6}$ is 17 ($p_{3,6} = (3 \rightarrow 2 \rightarrow 5 \rightarrow 6)$), and $d_{3,6}^-$ is 19 ($p_{3,6}^- = (3 \rightarrow 5 \rightarrow 6)$). Please note that these weights are perturbed weights after the perturbation of all non-betweenness and all-betweenness edges in Figure 4. After solving the inequalities, we see that t should be smaller than 2, and we select the largest rounded integer number 1. So $w_{2,5}^* = w_{2,5} + t = 5 + 1 = 6$.

Proposition 10. For a partial-betweenness edge $e_{i,j}$, we decrease its weight by t (the new perturbed weight is $w_{i,j}^* = w_{i,j} - t$) and t satisfies the following condition:

$$t < d_{s_1,i} + w_{i,j} + d_{j,s_2} - d_{s_1,s_2}, \quad (5)$$

for all p_{s_1,s_2} such that $e_{i,j} \notin p_{s_1,s_2}$,

then all p_{s_1,s_2}^* is not changed and some d_{s_1,s_2}^* is decreased ($p_{s_1,s_2}^* = p_{s_1,s_2}$).

The path which connects $p_{s_1,i}$, $e_{i,j}$ and p_{j,s_2} is the conditional shortest path between v_{s_1} and v_{s_2} through $e_{i,j}$. For

example, in Figure 6, the conditional shortest path between v_4 and v_6 through $e_{2,5}$ is $(4 \rightarrow 2 \rightarrow 5 \rightarrow 6)$, where $(4 \rightarrow 2)$ is the shortest path $p_{4,2}$, and $(5 \rightarrow 6)$ is the shortest path $p_{5,6}$. The meaning of Inequality (5) is that the weight of the conditional shortest path between s_1 and s_2 through $e_{i,j}$ should still be larger than the length of the perturbed path p_{s_1,s_2}^* .

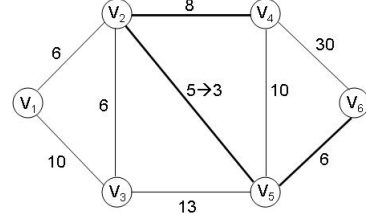


Fig. 6. Decreasing the weights of a partial-betweenness edge $e_{2,5}$

An example of decreasing the weight of a partial-betweenness edge is $e_{2,5}$ in Figure 6. The two targeted pairs in H , $p_{1,6}$ and $p_{3,6}$, do not pass through the edge $e_{2,5}$, but the shortest length path $p_{4,6}$ goes through it. Decreasing $w_{2,5}$ will not affect the shortest paths $p_{1,6}$ and $p_{3,6}$ but has something to do with $p_{4,6}$. Hence, there is only one constraint to decrease $w_{2,5}$ to $w_{2,5}^* = w_{2,5} - t$ as follows:

$$d_{4,2} + (w_{2,5} - t) + d_{5,6} > d_{4,6} \Rightarrow t < d_{4,2} + w_{2,5} + d_{5,6} - d_{4,6},$$

where $d_{4,2}$ is 8 ($p_{4,2} = (4 \rightarrow 2)$), $d_{5,6}$ is 6 ($p_{5,6} = (5 \rightarrow 6)$), and $d_{4,6}$ is 16 ($p_{4,6} = (4 \rightarrow 5 \rightarrow 6)$). After solving the inequality, we see that t should be smaller than 3, and we select the largest rounded integer number 2. So $w_{2,5}^* = w_{2,5} - t = 5 - 2 = 3$.

Summing up the above propositions briefly, a practical greedy perturbation process is as follows (the pseudocode is in Algorithm 1). Based on the adjacency weight matrix W , we first generate the shortest paths P and the corresponding lengths D by Floyd-Warshall algorithm [4] (see Line 1 of Algorithm 1). Then each edge $e_{i,j}$ in E is determined as one of the three categories: non-betweenness, all-betweenness or partial-betweenness. The non-betweenness edges and all-betweenness edges are perturbed based on Proposition 7 and Proposition 8 (see Line 2 and Line 3), respectively, before the partial-betweenness edges, and at the same time, the perturbed adjacency weight matrix W^* and the perturbed shortest path length matrix D^* are updated simultaneously. Then all partial-betweenness edges are sorted in a descending order of the number of the shortest paths which pass through this partial-betweenness edge. Such all partial-betweenness edges compose a stack PB. From the top to the bottom of this stack PB, we perturb the current top partial-betweenness edge $e_{i,j}$ by either Proposition 9 or Proposition 10 based on the verification whether the number of d_{s_1,s_2}^* ($e_{i,j} \in p_{s_1,s_2}$ and $d_{s_1,s_2}^* \leq d_{s_1,s_2}$) is larger than the number of d_{s_1,s_2}^* ($e_{i,j} \in p_{s_1,s_2}$ and $d_{s_1,s_2}^* > d_{s_1,s_2}$). If yes, the perturbed weight is increased according to Proposition 9 (see Lines 8-9). Otherwise, we decrease the weight based on Proposition 10 (see Lines 11-12). Observing these four propositions, we can know that all perturbed shortest paths will not be changed

in any case ($p_{s_1, s_2}^* = p_{s_1, s_2}$, for every (s_1, s_2) in H). The perturbed shortest path lengths will probably not be the same as the original ones ($d_{s_1, s_2}^* \neq d_{s_1, s_2}$), but the difference is minimized by the alternate choice of either weight increment or decrement.

Algorithm 1 Greedy Perturbation Algorithm

Input: The symmetric adjacency weight matrix W of an original graph G ; the set of selected shortest paths to be preserved H .

Output: The symmetric adjacency weight matrix W^* of the corresponding perturbed graph G^*

- 1: generate P and D based on W , and assign D to D^*
 - 2: for all non-betweenness edges $e_{i,j}$, $w_{i,j}^* \leftarrow w_{i,j} + r$ (r is any random positive number), and update D^*
 - 3: for all all-betweenness edges $e_{i,j}$, $w_{i,j}^* \leftarrow w_{i,j} - r$ (r is any random positive number which is smaller than $w_{i,j}$), and update D^*
 - 4: sort all partial-betweenness edges in a descending order with respect to the number of the shortest paths which pass through this partial-betweenness edge. Such all partial-betweenness edges compose a stack PB
 - 5: **while** PB $\neq \emptyset$ **do**
 - 6: pull out the top edge $e_{i,j}$ from PB
 - 7: **if** # of cases where $d_{s_1, s_2}^* \leq d_{s_1, s_2}$ is larger than # of cases where $d_{s_1, s_2}^* > d_{s_1, s_2}$ **then**
 - 8: generate a t based on Proposition 9
 - 9: $w_{i,j}^* \leftarrow w_{i,j} + t$
 - 10: **else**
 - 11: generate a t based on Proposition 10
 - 12: $w_{i,j}^* \leftarrow w_{i,j} - t$
 - 13: **end if**
 - 14: update D^*
 - 15: **end while**
-

IV. EXPERIMENTS

A. Databases

In the experiment section, we choose one real database, EIES Acquaintanceship at time 2, obtained from International Network for Social Network Analysis [9].

The Electronic Information Exchange System (EIES) data at time 2 were collected by Freeman and Freeman [9]. This dataset is discussed also in Wasserman and Faust [7]. This is a network of 48 researchers who participated in an early study on the effects of electronic information exchange, a precursor of email communication. The measure of acquaintanceship in this dataset has four levels, from 1 (do not know the other) to 4 (very good friendships). But the original dataset is nonsymmetric since the acquaintanceship in two people may not be the same. For example, A thinks B is his/her best friend, but B probably thinks A is a normal friend for him/her. To fit EIES data to our algorithms based on the undirected graph, we perform the following transformation to generate

our symmetric matrix W :

$$W_{i,j} = 9 - (E_{i,j} + E_{j,i}),$$

where $E_{i,j}$ is a numerical value in the range [1,4] which represents the i -th scholar original acquaintanceship to the j -th scholar.

In addition to the EIES database, to test the scalability of our greedy perturbation algorithm, we create a synthetic data which consists of 1600 objects and 70% objects are connected with each other, and the weights of the edges are randomly ranging from 10 to 100. Its corresponding adjacency weight matrix is a 1600*1600 symmetric matrix.

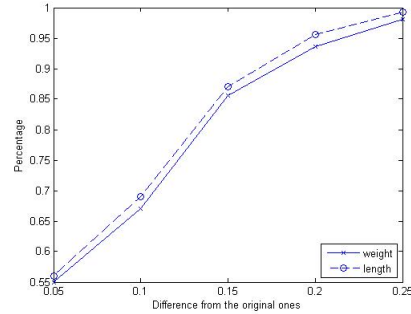


Fig. 7. Percentage of the preserved shortest path lengths and weights after the Gaussian perturbation with $\sigma=0.1$ on EIES

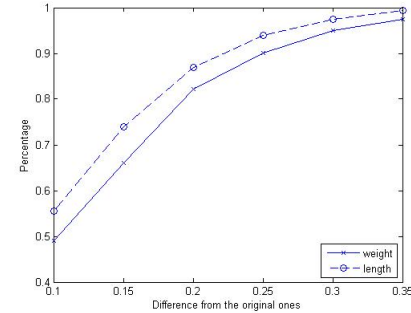


Fig. 8. Percentage of the preserved shortest path lengths and weights after the Gaussian perturbation with $\sigma=0.15$ on EIES

Figures 7, 8 and 9 show experimental results with different values of σ in Gaussian randomization multiplication. In each figure, the x -axis is the difference between the original ones and the corresponding preserved ones, and the y -axis denotes the percentage of either weights or lengths which fall within the x -axis difference. In each figure, there are two lines, a solid line and a dashed line. The dashed line represents the shortest path lengths and the solid line denotes the edge weights.

For example, in Figure 7, at x -axis 0.15, the dashed point (lengths) is 0.8699 and the solid point (weights) is 0.8565. It means that for each $w_{i,j}^* = w_{i,j}(1 - x_{i,j})$ ($x_{i,j}$ is from $N(0,0.1^2)$), 85.65% $w_{i,j}^*$ of the edges fall into $w_{i,j}(1 \pm 0.15)$, and 86.99% $d_{i,j}^*$ of the shortest path lengths fall into $d_{i,j}(1 \pm 0.15)$.

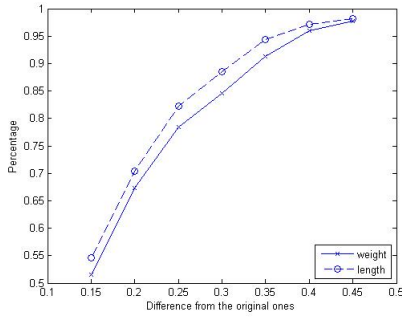


Fig. 9. Percentage of the preserved shortest path lengths and weights after the Gaussian perturbation with $\sigma=0.2$ on EIES

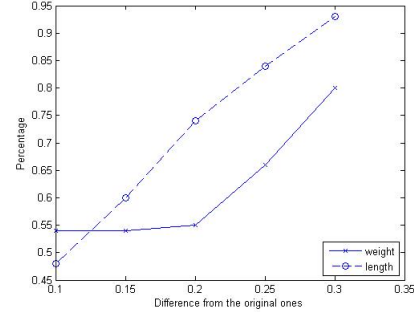
Based on Figures 7, 8 and 9, it is clear that the distribution of the shortest path lengths in the perturbed social network confirms the mathematical analysis in Section III-B: the percentage of the shortest path lengths in the perturbed social network which fall within $\pm\sigma$, $\pm 2\sigma$ and $\pm 3\sigma$ of those of the original social network is approximated 74%, 98% and 99%, respectively. In Figure 8 ($\sigma=0.15$), for example, at x -axis 0.15 ($0.15=\sigma$) the percentage of perturbed shortest path lengths close to the original one within σ is around 74%, at x -axis 0.3 ($0.3=2\sigma$) the percentage of perturbed shortest path lengths close to the original one within 2σ is around 98%. Figures 7 and 9 are also consistent with our mathematical analysis. More importantly, the percentage of difference between w^* and w is very close to the percentage of difference between d^* and d . As mentioned earlier, the Gaussian randomization multiplication strategy cannot guarantee the same shortest path preservation after the perturbation.

B. Results with Greedy Perturbation Algorithm

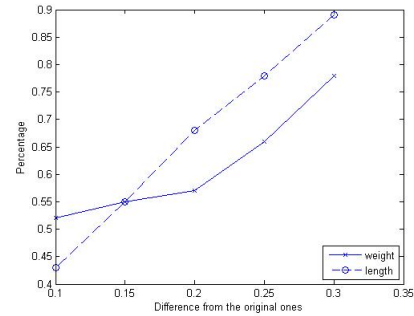
Before our greedy perturbation algorithm experiment, we point out that the weights of non-betweenness edges and all-betweenness could be changed dramatically without affecting any shortest paths in H . Hence, we only concern about the weights of all partial-betweenness edges in the two databases, EIES and synthetic data. Our experimental results with the greedy perturbation algorithm are shown in Figures 10, 11 and 12.

The explanation of these figures is that, for example, in Figure 10(a), at x -axis 0.15, the dashed line point (lengths) is 0.6 (60%) and the solid point (weights) is 0.54 (54%). It means that 54% $w_{i,j}^*$ of the edges fall into $w_{i,j}(1 \pm 0.15)$, and 60% $d_{i,j}^*$ of the shortest path lengths fall into $d_{i,j}(1 \pm 0.15)$, in addition to the shortest paths of all targeted nodes being exactly preserved.

Figures 10, 11 and 12 are three different experimental results based on various numbers of targeted pairs, 77%, 54%, 25%, which are what we want to keep exactly the same shortest paths and close shortest path lengths in the two databases. In other words, only 77%, 54% and 25% pairs of all pairs are included in the targeted pair set H , respectively. In addition to the various numbers of targeted pairs, the ratios



(a) EIES



(b) Synthetics

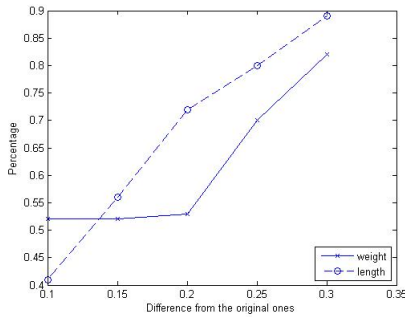
Fig. 10. Percentage of the preserved shortest path lengths and weights after the greedy perturbation with 77% targeted pairs being preserved

of partial-betweenness edges to all edges are 13%, 15% and 9% in EIES, and 19%, 14% and 20% in the synthetic data, respectively. For example, in Figure 10(a), the number of all edges is 820, but only 13% edges ($820 \times 13\% = 103$) are under the constraint while the other 87% edges could be changed dramatically and unconstrainedly.

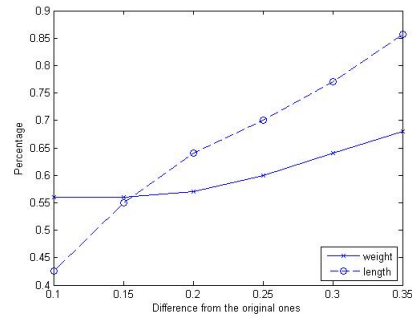
From Figures 10, 11 and 12, it is obvious that even a large amount of targeted pairs in H which need keep exactly the same shortest paths and close shortest path lengths, the perturbed shortest path lengths are still very close to the original ones. In addition to this, we should emphasize again that the shortest paths of all 77%, 54% and 25% targeted node pairs are exactly kept.

V. CONCLUSION AND FUTURE PLAN

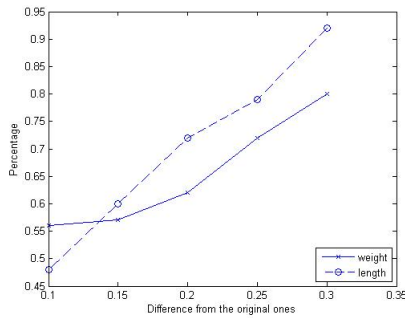
In consideration of the privacy issue in social network data mining techniques, the links between social network entities are sensitive in some cases such as the business transaction expenses, personal disease characteristic in an epidemiology mode, terrorist network relationship, and so forth. This paper addresses a balance between protection of sensitive weights of network links (edges) and some global structure utilities such as the shortest path length.



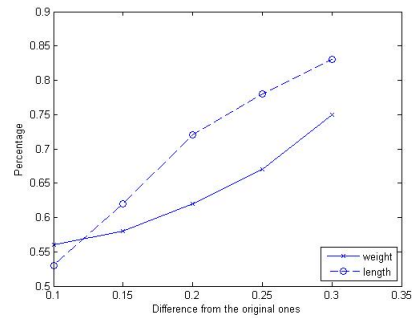
(a) EIES



(a) EIES



(b) Synthetics



(b) Synthetics

Fig. 11. Percentage of the preserved shortest path lengths and weights after the greedy perturbation with 54% targeted pairs being preserved

Fig. 12. Percentage of the preserved shortest path lengths and weights after the greedy perturbation with 25% targeted pairs being preserved

In this paper, we presented two perturbation strategies, Gaussian randomization multiplication and greedy perturbation algorithm to perturb individual (sensitive) edge weights and try to keep exactly the same shortest paths as well as their lengths close to those of the original social network. Our experimental results demonstrate that the two proposed perturbation strategies do meet the expectation of our mathematical analysis.

Further research work along this line can be carried out to extend our perturbation strategies to perturb the original edges in case of a dynamic evolutionary complex social network in which the social network structure and its weights change over time.

REFERENCES

- [1] L. Backstrom, C. Dwork, and J. Kleinberg, "Wherefore art thou R3579X? anonymized social networks, hidden patterns, and structural steganography," in Proceedings of the 16th international conference on World Wide Web, Alberta, Canada, pp. 181-190, 2007.
- [2] S. Bapna and A. Gangopadhyay, "A wavelet-based approach to preserve privacy for classification mining," *Decision Sciences Journal*, 37(4):623-642, 2006.
- [3] J. Baumes, M. Goldberg, M. Magdon-Ismael, and A. Wallace, "Discovering hidden groups in communication networks," in Proceedings of the 2nd NSF/NIJ Symposium on Intelligence and Security Informatics, Tucson, Arizona, pp. 378-389, June 2004.

- [4] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, 1st ed., the MIT Press, 1990.
- [5] P. C. Cross, J. O. Lloyd-Smith, and W. M. Getz, "Disentangling association patterns in fission-fusion societies using African buffalo as an example," *Animal Behaviour*, 69: 499-506, 2005.
- [6] A. Evfimievski, "Randomization in privacy preserving data mining," *ACM SIGKDD Explorations Newsletter*, 4(2):43-48, 2002.
- [7] K. Faust and S. Wasserman, "Social network analysis: methods and applications," Cambridge University Press, New York, NY, 1994.
- [8] I. R. Fischhoff, S. R. Sundaresan, J. Cordingley, H. M. Larkin, M. J. Sellier, and D. I. Rubenstein, "Social relationships and reproductive state influence leadership roles in movements of plains zebra, *Equus burchellii*," *Animal Behaviour*, 73(5): 825-831, 2007.
- [9] L. C. Freeman and S. C. Freeman, "A semi-visible college: structural effects on a social networks group," Henderson, M.M., and McNaughton, M.J. (eds.) *Electronic Communication: Technology and Impacts* Boulder, CO: Westview Press, pp. 77-85, 1980.
- [10] L. Getoor and C. P. Diehl, "Link mining: a survey," *ACM SIGKDD Explorations Newsletter*, 7(2): 3-12, 2005.
- [11] M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava, "Anonymizing social networks," University of Massachusetts, Amherst, MA, Tech. Rep. 07-19, 2007.
- [12] R. B. Jones, A. Gordus, J. A. Krall, and G. MacBeath, "A quantitative protein interaction network for the ErbB receptors using protein microarrays," *Nature*, 439(7073): 168-174, 2006.
- [13] L. Liu, J. Wang, Z. Lin, and J. Zhang, "Wavelet-based data distortion for privacy-preserving collaborative analysis," University of Kentucky, Lexington, KY, Tech. Rep. 482-07, July 2007.
- [14] L. A. Meyers, M. Newman, and B. Pourbohloul, "Predicting epidemics on directed contact networks," *Journal of Theoretical Biology*, 240: 400-418, 2006.

- [15] S. Mukherjee, Z. Chen, and A. Gangopadhyay, "A privacy preserving technique for Euclidean distance-based mining algorithms using Fourier-related transforms," *The VLDB Journal*, 15(4):293-315, 2006.
- [16] K. Muralidhar, R. Parsa, and R. Sarathy, "A general additive data perturbation method for database security," *Management Science*, 45(10): 1399-1415, 1999.
- [17] J. M. Read and M. J. Keeling, "Disease evolution on networks: the role of contact structure," *Proc. R. Soc. Lond. B*, 270: 699-708, 2003.
- [18] E. M. Rogers, *Diffusion of Innovations*, 5th ed., Simon & Shuster, Inc., 2003.
- [19] D. I. Rubenstein, S. Sundaresan, I. Fischhoff, and D. Saltz, "Social networks in wild asses: comparing patterns and processes among populations," *Erforsch. Biol. Ress. Mongolei (Halle/Saale)*, 10: 159-176, 2007.
- [20] S. M. Stigler, *Statistics on the Table*, Harvard University Press, 1999.
- [21] L. Sweeney, "Guaranteeing anonymity when sharing medical data, the DataFly system," *Journal of the American Medical Informatics Association*, Suppl. S, pp. 51-55, 1997.
- [22] S. Xu, J. Zhang, D. Han, and J. Wang, "Data distortion for privacy protection in a terrorist analysis system," in *Proceedings of the 2005 IEEE International Conference on Intelligence and Security Informatics*, Atlanta, GA, pp. 459-464, 2005.
- [23] S. Xu, J. Zhang, D. Han, and J. Wang, "Singular value decomposition based data distortion strategy for privacy protection," *Knowledge and Information Systems*, 10(3):383-397, 2006.
- [24] L. W. Young and R. B. Johnston, "The role of the internet in business-to-business network transformations: a novel case and theoretical analysis," *Information Systems and E-Business Management*, 1(1): 73-91, 2003.
- [25] E. Zheleva and L. Getoor, "Preserving the privacy of sensitive relationships in graph data," in *Proceedings of the First ACM SIGKDD International Workshop on Privacy, Security, and Trusting KDD*, San Jose, California, pp. 153-171, Aug 2007.
- [26] B. Zhou and J. Pei, "Preserving privacy in social networks against neighborhood attacks," in *Proceedings of the 24th International Conference on Data Engineering (ICDE'08)*, Cancun, Mexico, pp. 506-515, April 2008.