

# Privacy Preservation in Social Networks with Sensitive Edge Weights

Lian Liu \*

Jie Wang †

Jinze Liu ‡

Jun Zhang \*

## 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 a business transaction network, in which weights are attached to network edges that are considered to be confidential (e.g., transactions). We consider perturbing the weights of some edges to preserve data privacy when the network is published, while retaining the shortest path and the approximate cost of the path between some pairs of nodes in the original network. We develop two privacy-preserving strategies for this application. The first strategy is based on a Gaussian randomization multiplication, the second one is a greedy perturbation algorithm based on graph theory. In particular, the second strategy not only yields an approximate length of the shortest path while maintaining the shortest path between selected pairs of nodes, but also maximizes privacy preservation of the original weights. We present experimental results to support our mathematical analysis.

## 1 Introduction.

A social network is a special graph structure made of entities and connections between these entities. The 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 exchanges, friend relationships, conflict likelihood, web links, sexual relations, disease transmission (epidemiology), etc.

Social networks typically contain a large amount of private information. The need to protect confidential, sensi-

tive, and security information from being disclosed motivates us to develop privacy-preserving techniques for social networks. One of the major challenges, therefore, is to approach an optimal tradeoff between securing the confidential information and maximizing the social network's utility analysis.

Recent study of privacy preservation in social networks focuses on the de-identification process to protect the privacy of individuals while preserving the patterns between small communities [6, 19, 20]. Such de-identification processes are often helpful when the individual's identification is considered to be confidential, such as a patient's identity.

However, the individual identity is not always considered to be confidential. For example, a recent tool called ArnetMiner [15] has been developed to allow mining the academic research network through a public web portal. Each node of this network represents a researcher. An edge exists between two nodes if the corresponding researchers share the co-authorship. Another feature that is supported by the system is the association search between two researchers, which enumerates all possible topics that connect one researcher to the other and how closely two researchers are connected. In this case, since all data needed to compute such network are obtained from public web pages or databases, privacy is not a big concern. However, it is important to realize that the network derived from these public data makes implicit knowledge explicit and more specific, such as the association between individuals.

Next, we give another example of weighted social networks, which is thoroughly studied in [7]. The social network represents an automotive business network between Japanese corporations and American suppliers in North America. The background behind this example is that many Japanese automotive companies have already taken roots in North America, and it is of interest to American suppliers to seek access to such a profitable subcontract market. On one hand, the existence of a long-term and loyal connection between Japanese first-tier suppliers and themselves plays a key role in making decisions. So these preferences surely prevent American suppliers from obtaining contracts. On the other hand, since most first-tier suppliers are sensitive to importing cost and have U.S. political pressure to avoid mass outsourcing, they prefer to collaborate with the qualified local American suppliers. Therefore, it is practical and economical to become a subcontractor of these lower-level

\*Laboratory for High Performance Scientific Computing and Computer Simulation, Department of Computer Science, University of Kentucky, Lexington, KY 40506-0046, USA. Email: lliuc@csr.uky.edu, jzhang@cs.uky.edu

†Computer Science Department, Minnesota State University Mankato, 273 Wissink Hall, Mankato, MN 56001, USA. Email: jie.wang@mnsu.edu

‡Department of Computer Science, College of Engineering, University of Kentucky, 237 Hardyman, Lexington, KY, 40506-0046, USA. Email: liuj@cs.uky.edu

suppliers. For every potential American supply contractor, it is desirable to obtain a comprehensive business network that can guide them in finding the most economical business path.

However, due to the fierce competition between suppliers, managers may not be willing to disclose the true transaction expenses to their adversaries. Otherwise, their adversaries probably reduce the quotation below the price obtained in a secret bidding competition. Hence, suppliers would like to preserve their transaction expenses (edge weights) before the business network is published. At the same time, some global and local utilities of social networks, such as the optimal supply chains (the lowest cost path between companies) and the corresponding lengths, are probably desired to be maintained for future analysis.

In this paper, we focus on publishing a social network which maintains the utility of the shortest paths while perturbing the actual cost between a pair of entities. The edge between two nodes is often associated with a quantitative weight that reflects the affinity between the two entities. The weighted graph allows deeper understanding about relationships between entities within the network. The shortest path between a pair of nodes is a path such that the sum of the weights of its constituent edges is minimized. The shortest path is a major data utility which has applications in different fields.

So each node in this business graph represents a company or a supplier (we call it an agent), the edge denotes business relationship and the weight of the edge represents the transaction expenses according to some measures (such as per month, per person or per transaction) between the two entities [18]. As an abstract business network in Figure 1, the bold numbers beside edges are the transaction expenses per month (the unit is million/month). In our business example, for example, Company A wants to purchase some products or services, in the future, from Company D which cannot directly access each other due to some trade barriers. Company A needs to choose some trade intermediate suppliers who have the most competitive path (the shortest path of price) between themselves and Company D (maybe these suppliers need other suppliers to connect Company D). If the weights of the business social network are perturbed as in Figure 2 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-preserving social network without having to know confidential details of the relationship between agents and Company D.

According to our proposed algorithms, the perturbed graph preserves the same shortest paths and maintains the shortest path lengths close to the true values. Moreover, the total privacy of all edge weights is maximized by our methods. As the example in Figure 1, the true expense between Agent 2 (or Supplier 2) and Company D is lower

than that between Agent 3 (or Supplier 3) and Company D, but in the perturbed network as in Figure 2, the expense between Agent 2 and Company D is higher than that between Agent 3 and Company D. So in a bidding competition, the business secret between Agent 2 and Company D is blind to Agent 3 (Agent 2's adversary) even if the perturbed business network is published. After a series of perturbations, the final perturbed version is in Figure 2. The shortest path between Company A and Company D is the same as the original one and the corresponding perturbed length is close to the original one.

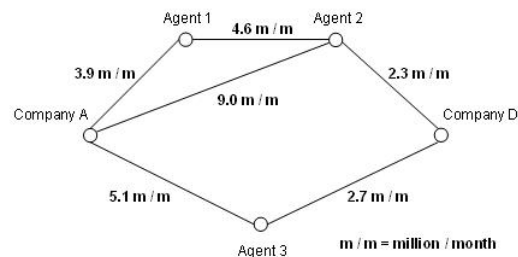


Figure 1: Original business network. All nodes in this figure represent either a company or an agent (supplier) and the edge means a business connection between the two entities. The weight of each edge denotes the transaction expense of the corresponding business connection.

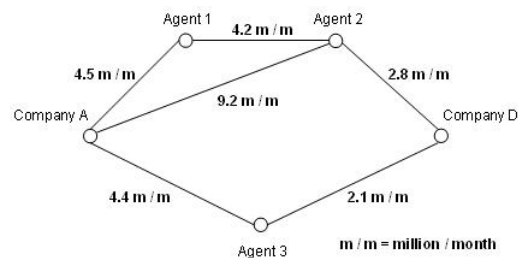


Figure 2: Perturbed business network.

To utilize the privacy-preserving social network analysis, each person (or organization) has a local (private) weighted graph before perturbation. The process of information sharing and perturbing can be done either in a distributed environment or a central situation. In a distributed environment, each person perturbs the individual local weighted graph, and then publishes the perturbed weights to the public. After all people's perturbation and publication, a global perturbed graph will be composed of individual's local perturbed graphs. In a central case, we assume there exists a trusted third-party which will absolutely never collude with anyone. Each person submits the original graph structure along with edge's weights to the trusted third-party which then perturbs the whole graph with the aid of our analysis

algorithms. After the central perturbation, the third-party releases the perturbed social network to the public.

Although just revealing the shortest paths and hiding all weights of edges between any two nodes can achieve privacy preservation in some cases, the unweighted shortest paths cannot achieve the same utility as the weighted ones in a real world. For example, in Figure 1, if we hide all weights and only show Company A that (Agent 1→Agent 2→Company D), (Agent 3→Company D) are the shortest paths between Agent 1 and Company D, and Agent 3 and Company D, respectively, Company A cannot choose an optimal one between the two paths to Company D just based on the unweighted shortest paths. In this unweighted graph, the two unweighted shortest paths are equivalent to some extent, but actually they are essentially different for Company A since the shortest path (Agent 3→Company D) is shorter (and more economical) than (Agent 1→Agent 2→Company D). Therefore, we need to preserve the shortest paths as well as the corresponding shortest path's lengths which facilitate business decision-making in a competitive environment.

So, in this paper, we consider perturbing edge weights while trying to preserve the shortest paths between pairs of nodes without adding or deleting any node and edge. For this purpose, we propose two perturbation strategies, Gaussian randomization multiplication and greedy perturbation. The two strategies serve different purposes. The Gaussian method mainly focuses on preserving the lengths of the perturbed shortest paths within some bounds of the original ones but does not guarantee the same shortest path after perturbation. The advantages of the greedy perturbation algorithm over the Gaussian algorithm are that it can keep the same shortest paths during the perturbation, in addition to keeping the perturbed shortest path lengths close to those of the original ones.

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 2. Two edge privacy-preserving strategies and theoretical analyses are presented in Section 3. Experimental results are listed and discussed in Section 4. Finally a brief conclusion is given in Section 5.

## 2 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. Meth-

ods 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 [1, 8, 9, 16, 17] and randomization addition [3, 10].

In social networks, the data is not meaningfully represented by a tabular or matrix. Hence, most people do not use traditional matrix-based algorithms to preserve privacy. They emphasize the protection of social entity's identification via de-identification techniques [14]. For example, Hay et al. [6] and Zhou et al. [20] 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. [11] and Rogers [12] 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. [19] 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.

These 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.

## 3 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 others 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.

We firstly give some notations that will be used later, and then introduce our two strategies, Gaussian randomization multiplication and greedy perturbation algorithm.

**3.1 Preliminaries and Notations.** A social network in this paper is defined as an undirected and weighted graph

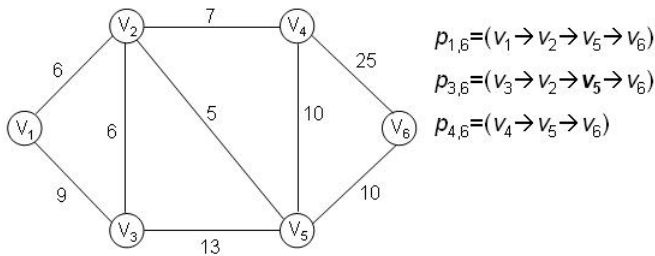


Figure 3: A simple social network  $G$  and the three shortest paths.

$G = \{V, E, W\}$ . Figure 3 is a simple social network. The nodes of the graph,  $V$ , may denote meaningful entities from the real world such as individuals, organs, organizations, communities, and so on. In Figure 3,  $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ .  $E$  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 3. All  $w_{i,j}$  form 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 for the 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, respectively.

**3.2 Distributed Perturbation by 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.

The basic idea behind this algorithm is that every two linked entities cooperate with the generation of a random number which is consistent with a Gaussian distribution. The weight of the edge connecting these two entities is multiplied by the random number and the individual perturbed weight is released to the public. Because each edge's random number and the edge's perturbation process is only related to these two linked entities, the random number generation and weight perturbation have nothing to do with other edges. In other words, the perturbation of all edge's weights can be done in a distributed environment. The maximum increment

or decrement of each weight is only dependent on the parameters of this distribution. So the shortest paths and corresponding lengths will probably be preserved if the parameters of the Gaussian distribution are chosen appropriately. We assume that the parameters of the Gaussian distribution are predefined and globally known.

**Proposition 1.** *There does not exist a perturbation schema such that every edge weight is perturbed but the shortest paths and the corresponding lengths 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, after the perturbation, 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 [2]), and  $d_{i,k_h} = d_{i,k_h}^*$ . It follows that

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

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

**Gaussian randomization multiplication strategy.** We assume that  $W$  is an  $n * n$  matrix whose entries are either weights if two nodes have a link or  $\infty$  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  $n * n$  symmetric Gaussian noise matrix with the mean 0 and the standard deviation  $\sigma$ . 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  $N(0, \sigma^2)$ . If node  $v_i$  has a connection with  $v_j$ , then  $v_i$  generates a random number,  $x_{i,j}^1$ , from the Gaussian distribution  $N(0, \sigma^2)$ , and  $v_j$  also generates a random number,  $x_{i,j}^2$ , from the same distribution.  $x_{i,j}$  could be the averaged value between  $x_{i,j}^1$  and  $x_{i,j}^2$ . The Gaussian-perturbed version of the graph  $G$  in Figure 3 is shown in Figure 4. Here, the symmetric Gaussian noise matrix is generated from  $N(0, 0.15^2)$  ( $\sigma=0.15$ ).

Note that the above multiplication is based on undirected graphs. If we need to extend it to directed graph cases,

the cooperation of generating  $x_{i,j}$  is not necessary. Instead, if node  $v_i$  has a directed edge from node  $i$  to node  $j$ , then node  $i$  can directly generate a random number  $x_{i,j}$  from the Gaussian distribution without the cooperation with node  $j$ . Other procedures are same as the above undirected graph case.

The reasons why we chose the Gaussian randomization multiplication strategy are as follows. 1). It is straightforward 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 will emerge in the future 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 useless 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.

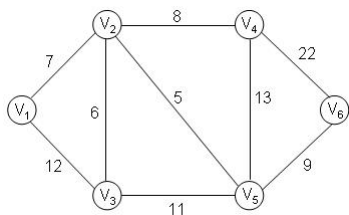


Figure 4: The perturbed social network  $G^*$  of  $G$  in Figure 3. Compared to Figure 3, all weights in this figure except  $w_{2,3}$  and  $w_{2,5}$  are perturbed.

In Figure 4, all values of  $V^*$  and  $E^*$  are same as those of  $V$  and  $E$  in Figure 3. The major 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,  $2\sigma$ , as shown in Theorem 2.

**Theorem 2.** *In the Gaussian randomization multiplication strategy, we assume the length of a path ( $v_i \rightarrow v_{k_1} \rightarrow v_{k_2} \rightarrow \dots \rightarrow v_{k_h} \rightarrow v_j$ ) is  $L_{i,j}$  (their edges are  $e_{i,k_1}, e_{k_1,k_2}, \dots, e_{k_{h-1},k_h}, e_{k_h,j}$ , and their weights are  $w_{i,k_1}, w_{k_1,k_2}, \dots, w_{k_h,j}$ ).  $L_{i,j}^*$  and  $w_{i,k_1}^*, w_{k_1,k_2}^*, \dots, w_{k_h,j}^*$  are the perturbed values after the Gaussian algorithm, then*

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

where  $i$  and  $j$  denote the beginning and ending nodes of the path,  $\sigma$  is the standard deviation of the Gaussian distribution and  $n$  can be any positive integer.

*Proof.*  $\Pr\left(\frac{|L_{i,j}^* - L_{i,j}|}{L_{i,j}} \leq n\sigma\right)$  is the probability function of  $\frac{|L_{i,j}^* - L_{i,j}|}{L_{i,j}}$  being smaller than  $n\sigma$ .  $\operatorname{erf}(\Delta)$  is the Gaussian error function.  $L_{i,j} = w_{i,k_1} + w_{k_1,k_2} + \dots + w_{k_h,j}$ , 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

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

Sum up the above equations,

$$\begin{aligned} L_{i,j}^* &\geq L_{i,j}(1 - u), \\ (3.1) \quad \frac{|L_{i,j}^* - L_{i,j}|}{L_{i,j}} &\leq u. \end{aligned}$$

Take the probability function on both sides of Inequality (3.1), we obtain

$$(3.2) \quad \Pr\left(\frac{|L_{i,j}^* - L_{i,j}|}{L_{i,j}} \leq n\sigma\right) \geq \Pr(u \leq n\sigma).$$

According to [13], in a Gaussian distribution ( $u$  is the maximum value of the absolute numbers generated from a Gaussian distribution),  $\Pr(u \leq n\sigma) \geq \operatorname{erf}\left(\frac{n}{\sqrt{2}}\right)$ . So, Inequality (3.2) extends to:

$$\begin{aligned} \Pr\left(\frac{|L_{i,j}^* - L_{i,j}|}{L_{i,j}} \leq n\sigma\right) &\geq \Pr(u \leq n\sigma) \\ (3.3) \quad &\geq \operatorname{erf}\left(\frac{n}{\sqrt{2}}\right). \end{aligned}$$

□

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

From [13], we can easily see 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  $\sigma$ , 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 protecting the exact edge weights of the original networks from disclosure.

Comparing Figure 3 to Figure 4, 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  $2\sigma$ . 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 the 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 the perturbed shortest path lengths falling within the range  $\pm 2\sigma$  is  $14/15=93\%$  which is consistent with our mathematical analysis in Theorem 2.

**Corollary 3.** *Let  $d_{i,j}$  be the length of the shortest path between node  $i$  and  $j$ . We assume  $d_{i,j}^{second}$  is the length of the second shortest path between them. We define a ratio*

$$\beta_{i,j} = \frac{d_{i,j}^{second} - d_{i,j}}{d_{i,j}}.$$

*If  $\beta_{i,j}$  is greater than  $2\sigma$ , the shortest path is highly possible to be preserved after our Gaussian randomization multiplication strategy. Here,  $\sigma$  is the parameter of the Gaussian noise matrix  $N(0, \sigma^2)$ .*

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

Comparing Figure 3 to Figure 4 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 an equal length, ( $p_{3,5}^*=(v_3 \rightarrow v_5)$  or  $(v_3 \rightarrow v_2 \rightarrow v_5)$ ),  $p_{4,5}^*=(v_4 \rightarrow v_5)$  or  $(v_4 \rightarrow v_2 \rightarrow v_5)$ ,  $p_{4,6}^*=(v_4 \rightarrow v_6)$  or  $(v_4 \rightarrow v_5 \rightarrow v_6)$ ), the second of these is different from the corresponding original ones. Therefore we consider that their perturbed shortest paths are changed even one of their perturbed shortest paths is the same as that of the original one.

But the Gaussian randomization multiplication strategy cannot guarantee the same shortest path preservation after perturbation, if  $\beta_{i,j}$  is very small. For example, the original shortest path length between  $v_3$  and  $v_5$  in Figure 3 is 11 ( $v_3 \rightarrow v_2 \rightarrow v_5$ ) and the original second shortest path length is 13 ( $v_3 \rightarrow v_5$ ). Its ratio  $\beta_{3,5}$  is  $(13-11)/11=0.18$  which is not greater than  $2\sigma$ . According to Corollary 3, the perturbed shortest path may be changed after the Gaussian strategy. Actually, in our example,  $p_{3,5}^*$  has two different shortest paths which are not considered to be 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 3 is 21 ( $v_1 \rightarrow v_2 \rightarrow v_5 \rightarrow v_6$ ) and the original second shortest path length is 30 ( $v_1 \rightarrow v_3 \rightarrow v_2 \rightarrow v_5 \rightarrow v_6$ ). So the perturbed shortest path,  $p_{1,6}^*$ , is exactly preserved since the ratio is  $(30-21)/21=0.43$  which is greater than  $2\sigma$ .

Therefore, we give another strategy to ensure that, for the 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.

### 3.3 Shortest Path Preserving 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. But we need a trusted third-party who will absolutely never collude with any network entities. All social network entities submit their original graph structures along with the edge's weights to the third-party. Then all analysis and perturbation procedures are done by the third-party, and a global perturbed social network will be published to the public after the perturbation. Because all analysis and perturbation are done by a central third-party, the undirected social network and directed one have a very similar procedure. In detail, only the directed edges (and the corresponding weights) and directed paths (and the corresponding lengths) are chosen to be fed into the following analysis and perturbation in a directed social network. So, we do not distinguish the difference between undirected and directed social networks below.

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. Actually, in the real world, it is not reasonable that all information is considered as confidential. We further assume that only the data owner has the right to select which shortest path should be preserved or which one should not be preserved. Our tasks are, under data owner's restrictions, to maximize the preservation of edge weight's privacy and minimize the difference of the shortest paths and the corresponding lengths between the original social networks and perturbed ones as much as possible.

In other words, we want to keep parts of the shortest paths (the starting and ending nodes,  $(s_1, s_2)$ , in the shortest paths compose a node pair set  $H$ , see below) and the corresponding lengths as close to the original ones as possible, while ignoring 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 much as possible. For example, in the graph  $G=\{V, E, W\}$  in Figure 3, let  $H$  be  $\{(1,6), (4,6), (3,6)\}$ . In a real social network, some of the 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 the 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 the shortest path list set  $P$  and the corresponding length  $n * n$  matrix  $D$ . In  $P$ , each entry  $p_{s_1, s_2}$  is a linked list representing the shortest path between  $s_1$  and  $s_2$ , (i.e.,  $s_1$  and  $s_2$  is the beginning and ending nodes of the shortest path, respectively). For example,  $p_{1,6}=(v_1 \rightarrow v_2 \rightarrow v_5 \rightarrow v_6)$ , 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  $s_1$  and  $s_2$ . 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 conditions in Figure 5.

1.  $V^* = V$  and  $E^* = E$ ,
  2. maximize the number of  $w_{i,j}^*$  such that  $w_{i,j}^* \neq w_{i,j}$ ,
  3.  $d_{s_1, s_2}^* \approx d_{s_1, s_2}$ , for every  $(s_1, s_2)$  in  $H$ ,
  4.  $p_{s_1, s_2}^* = p_{s_1, s_2}$ , for every  $(s_1, s_2)$  in  $H$ .
- Here,  $s_1$  and  $s_2$  are the beginning and ending nodes of the shortest paths in  $H$ , respectively.

Figure 5: The formulization of our perturbation purposes.

Based on the combination of the above conditions and the collected information, like  $P$  and  $D$ , we divide all edges in  $G$  into three different categories as in Figure 6 based on their involvement in the shortest paths to be preserved.

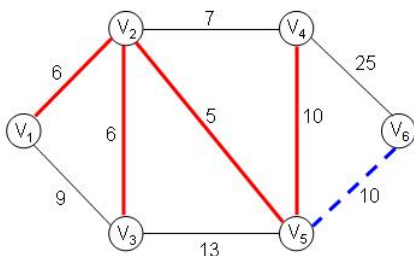


Figure 6: Three different categories of edges. The red bold-faced edges are partially-visited edges, the black thin edges are non-visited ones, and the blue dashed edge is the all-visited edge.

**Definition 4.** An edge  $e_{i,j}$  is a non-visited edge, if  $e_{i,j} \notin p_{s_1, s_2}$  for every  $(s_1, s_2) \in H$ . In other words, none of the shortest path in  $P$  passes through the edge  $e_{i,j}$ .

In Figure 6, all black thin edges such as edges  $e_{1,3}$ ,  $e_{2,4}$ ,  $e_{4,6}$  and  $e_{3,5}$  are non-visited edges, because the shortest 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-visited edges are the majority of edges in a social network.

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

In Figure 6, the blue dashed edge  $e_{5,6}$  is the all-visited edge since the shortest paths  $p_{1,6}$ ,  $p_{4,6}$  and  $p_{3,6}$  in  $H$  all go

through the edge  $e_{5,6}$ . Typically, the all-visited edges are very rare in a real social network.

**Definition 6.** An edge  $e_{i,j}$  is a partially-visited edge, if  $\exists (s_1, s_2) \in H$  and  $\exists (s_3, s_4) \in H$  such that  $e_{i,j} \in p_{s_1, s_2}$ , but  $e_{i,j} \notin p_{s_3, s_4}$ . 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 red bold-faced edges in Figure 6 are the partially-visited edges. For example,  $e_{2,5}$  is a partially-visited 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 four different schemes based on these three different categories.

**Proposition 7.** For a non-visited 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  $p_{s_1, s_2}^* = p_{s_1, s_2}$ ).

Because nobody in  $H$  passes any non-visited edge, increasing the weights of non-visited edges to any value will not change the shortest paths and the corresponding lengths in  $H$ .

**Proposition 8.** For an all-visited 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  $d_{s_1, s_2}^* = d_{s_1, s_2} - t$ .

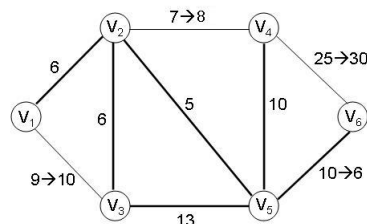


Figure 7: Perturbation on the non-visited and all-visited edges.

As in the social network shown in Figure 3, we perturb the non-visited and all-visited edges as in Figure 7. We increase the weights of the non-visited edges  $e_{1,3}$ ,  $e_{2,4}$  and  $e_{4,6}$ , and decrease the weight of the all-visited edge  $e_{5,6}$ .

In a social network, partially-visited edges are prevalent which are our major perturbation targets. To minimize the difference between the length of the original shortest path and that of the corresponding perturbed shortest path, we develop two perturbation schemes on partially-visited edges. If the current length of the perturbed shortest path is bigger than the original one, we can decrease the weight of one edge in this path. Otherwise, we can increase its weight. So

increasing and decreasing are two alternate choices to keep the length of the perturbed shortest path close to the original one.

**Proposition 9.** For a partially-visited 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:

$$0 < t < \min\{d_{s_1,s_2}^- - d_{s_1,s_2} \mid \text{for all } p_{s_1,s_2} \text{ such that } e_{i,j} \in p_{s_1,s_2}\},$$

where  $d_{s_1,s_2}^-$  is the length of the conditional shortest path between node  $s_1$  and node  $s_2$  in a graph  $G^- = \{V, E - \{e_{i,j}, e_{j,i}\}, W - \{w_{i,j}, w_{j,i}\}\}$ .  $G^-$  is the graph in which we only delete the edges  $e_{i,j}$  and  $e_{j,i}$  and the corresponding weights from  $G$ . 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  $e_{i,j}$  is in  $p_{s_1,s_2}$ ) will become larger, ( $p_{s_1,s_2}^* = p_{s_1,s_2}$  and  $d_{s_1,s_2}^* = d_{s_1,s_2} + t$ ).

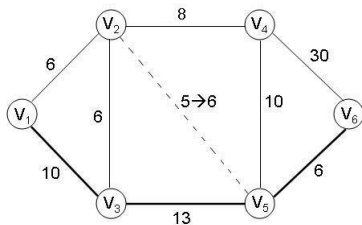


Figure 8: Increasing the weight of the partially-visited edge  $e_{2,5}$ .

An example of increasing the weight of the partially-visited edge  $e_{2,5}$  is shown in Figure 8. The shortest paths of 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 ( $p_{1,6} = (v_1 \rightarrow v_2 \rightarrow v_5 \rightarrow v_6)$ ),  $d_{1,6}^-$  is 29 ( $p_{1,6}^- = (v_1 \rightarrow v_3 \rightarrow v_5 \rightarrow v_6)$ ),  $d_{3,6}$  is 17 ( $p_{3,6} = (v_3 \rightarrow v_2 \rightarrow v_5 \rightarrow v_6)$ ), and  $d_{3,6}^-$  is 19 ( $p_{3,6}^- = (v_3 \rightarrow v_5 \rightarrow v_6)$ ). Note that these weights are perturbed weights after the perturbation of all non-visited and all-visited edges as shown in Figure 7. 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 partially-visited 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:

$$(3.4) 0 < t < \min\{d_{s_1,i} + w_{i,j} + d_{j,s_2} - d_{s_1,s_2} \mid \text{for all } p_{s_1,s_2} \text{ such that } e_{i,j} \notin p_{s_1,s_2}\},$$

then all  $p_{s_1,s_2}^*$  is not changed and some  $d_{s_1,s_2}^* = d_{s_1,s_2} - t$  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  $s_1$  and  $s_2$  through  $e_{i,j}$ . For example, in Figure 9, the conditional shortest path between  $v_4$  and  $v_6$  through  $e_{2,5}$  is  $(v_4 \rightarrow v_2 \rightarrow v_5 \rightarrow v_6)$ , where  $(v_4 \rightarrow v_2)$  is the shortest path  $p_{4,2}$ , and  $(v_5 \rightarrow v_6)$  is the shortest path  $p_{5,6}$ . The meaning of Inequality (3.4) is that the length 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  $p_{s_1,s_2}^*$ .

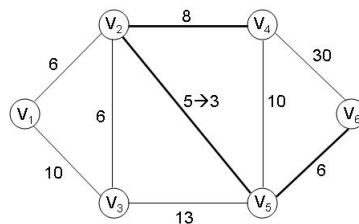


Figure 9: Decreasing the weight of a partially-visited edge  $e_{2,5}$ .

An example of decreasing the weight of the partially-visited edge  $e_{2,5}$  is depicted in Figure 9. The shortest paths of 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. 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 ( $p_{4,2} = (v_4 \rightarrow v_2)$ ),  $d_{5,6}$  is 6 ( $p_{5,6} = (v_5 \rightarrow v_6)$ ), and  $d_{4,6}$  is 16 ( $p_{4,6} = (v_4 \rightarrow v_5 \rightarrow v_6)$ ). 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 aforementioned propositions briefly, a practical greedy perturbation process is as follows (the pseudocode is in Algorithm 1). Based on the original adjacency weight matrix  $W$ , we first generate the shortest paths  $P$  and the corresponding lengths  $D$  by Floyd-Warshall algorithm [2] (see Line 1 of Algorithm 1). Then each edge  $e_{i,j}$  in  $E$  is determined as in one of the three categories: non-visited, all-visited or partially-visited. The non-visited edges

and all-visited edges are perturbed based on Proposition 7 and Proposition 8 (see Line 2 and Line 3), respectively, before the partially-visited 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 partially-visited edges are sorted in a descending order based on the number of the shortest paths passing through this partially-visited edge. Such all partially-visited edges form a stack PB. From the top to the bottom of this stack PB, we pop out the current top partially-visited edge  $e_{i,j}$ , and perturb  $e_{i,j}$  only once 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  $d_{s_1,s_2}^* \leq$  the original one) is larger than the number of  $d_{s_1,s_2}^*$  ( $e_{i,j} \in p_{s_1,s_2}$  and  $d_{s_1,s_2}^* >$  the original one). 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). Note that an edge popped out from PB will never be put back in the stack again. In other words, every partially-visited edge is perturbed only once and the perturbation is a one pass procedure. After perturbing the weight of any edge, we will recalculate and update the lengths of the all-pair shortest paths in  $D^*$  by Floyd-Warshall algorithm. According to these four propositions, we know that all the 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  $H$  according to Propositions 9 and 10). 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.

## 4 Experiments.

**4.1 Databases.** In the experiment section, we chose one real database, EIES (Electronic Information Exchange System) Acquaintanceship at time 2, obtained from International Network for Social Network Analysis [5].

The EIES data at time 2 were collected by Freeman and Freeman [5]. This dataset was also discussed in Wasserman and Faust [4]. 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). 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. Therefore, the social network in this dataset is directed and weighted.

In addition to the EIES database, to test the scalability of our greedy perturbation algorithm, we created a synthetic database which consists of 1600 objects and 70% objects are connected with each other, and the weights of the edges range randomly from 10 to 100. Its corresponding adjacency

---

### Algorithm 1 Greedy Perturbation Algorithm.

---

**Input:** The symmetric adjacency weight matrix  $W$  of an original graph  $G$  and  $H$  (the set of selected shortest paths to be preserved).

**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-visited 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-visited 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 partially-visited edges in a descending order with respect to the number of the shortest paths which pass through this partially-visited edge. Such all partially-visited edges form a stack PB
  - 5: **while** PB  $\neq \emptyset$  **do**
  - 6:   pop out the top edge  $e_{i,j}$  from PB
  - 7:   **if** # of cases where  $d_{s_1,s_2}^* \leq$  the original one is larger than # of cases where  $d_{s_1,s_2}^* >$  the original one **then**
  - 8:     generate a random value  $t$  given the range determined by Proposition 9
  - 9:      $w_{i,j}^* \leftarrow w_{i,j} + t$
  - 10:   **else**
  - 11:     generate a random value  $t$  given the range determined by Proposition 10
  - 12:      $w_{i,j}^* \leftarrow w_{i,j} - t$
  - 13:   **end if**
  - 14:   update  $D^*$
  - 15: **end while**
- 

weight matrix is a 1600\*1600 symmetric matrix.

**4.2 Results with Gaussian Randomization Multiplication Algorithm.** Figures 10, 11 and 12 show experimental results with different values of  $\sigma$  in Gaussian randomization multiplication. In each figure, the  $x$ -axis is the difference between the original ones and the corresponding perturbed ones, and the  $y$ -axis denotes the percentage of either perturbed weights or perturbed lengths which fall within the  $x$ -axis difference to the original ones. In each figure, there are two lines, a dashed line and a solid line. The dashed line represents the perturbed shortest path lengths and the solid line denotes the perturbed edge weights.

For example, in Figure 10, at  $x$ -axis 0.15, the dashed point (length) is 0.8699 and the solid point (weight) is 0.8565. It means that, in the Gaussian algorithm, 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 perturbed edges fall into  $w_{i,j}(1 \pm 0.15)$ , and 86.99%  $d_{i,j}^*$  of the perturbed shortest paths fall into  $d_{i,j}(1 \pm 0.15)$ .

Based on Figures 10, 11 and 12, it is clear that the distribution of the shortest path lengths in the perturbed

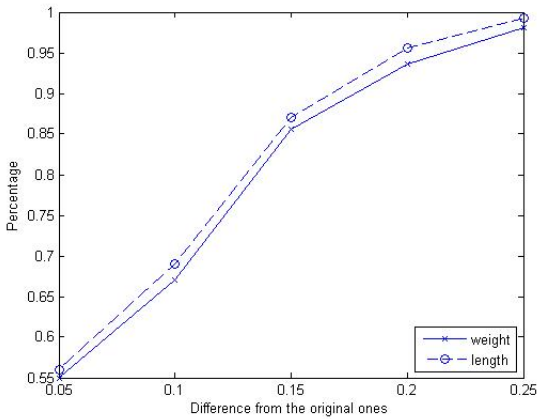


Figure 10: Percentage of the perturbed shortest path lengths and weights in the range after the Gaussian perturbation with  $\sigma=0.1$  on EIES.

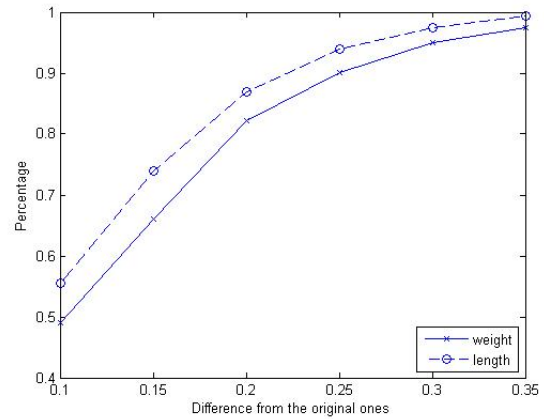


Figure 11: Percentage of the perturbed shortest path lengths and weights in the range after the Gaussian perturbation with  $\sigma=0.15$  on EIES.

social network confirms the mathematical analysis in Section 3.2: 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 approximately 68%, 95% and 99%, respectively. In Figure 11 ( $\sigma=0.15$ ), for example, at  $x$ -axis 0.15 ( $0.15=\sigma$ ) the percentage of the perturbed shortest path lengths close to the original ones within  $\pm\sigma$  is around 74%; at  $x$ -axis 0.3 ( $0.3=2\sigma$ ) the percentage of the perturbed shortest path lengths close to the original ones within  $\pm 2\sigma$  is around 98%. Figures 10 and 12 are also consistent with this mathematical analysis. More importantly, the percentage of difference between  $w^*$  and  $w$  is very close to the percentage of difference between  $d^*$  and  $d$ , (in these three figures, the two lines are similar to each other at all  $x$ -axis points). As mentioned earlier, however, the Gaussian randomization multiplication strategy cannot guarantee the same shortest path preservation after the perturbation.

**4.3 Results with Greedy Perturbation Algorithm.** Before our greedy perturbation algorithm experiment, we point out that the weights of non-visited edges and all-visited edges could be changed dramatically without affecting any of the shortest paths in  $H$ . Hence, we only concerned about the weights of all partially-visited edges in the two databases, EIES and synthetic data. Our experimental results with the greedy perturbation algorithm are shown in Figures 13, 14 and 15.

The interpretation of these figures is that, for example, in Figure 13(a), at  $x$ -axis 0.15, the dashed line point (length) is 0.6 (60%) and the solid point (weight) is 0.54 (54%). It means that, after the greedy perturbation algorithm, 54%  $w_{i,j}^*$  of the perturbed edges fall into  $w_{i,j}(1 \pm 0.15)$ , and

60%  $d_{i,j}^*$  of the perturbed shortest path lengths fall into  $d_{i,j}(1 \pm 0.15)$ , in addition to the shortest paths of all targeted pairs in  $H$  being exactly preserved.

Figures 13, 14 and 15 are three different experimental results based on various numbers of targeted pairs, 77%, 54%, 25%, which are what we wanted to keep exactly the same shortest paths and the close lengths of the shortest paths in the two databases. In other words, only 77%, 54% and 25% pairs of all pairs were included in the targeted pair set  $H$ , respectively. In addition to the various numbers of targeted pairs, the ratios of partially-visited 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 13(a), the number of all edges is 820, but only 13% edges ( $820 \times 13\% = 103$ ) are partially-visited edges and under the constraint while the other 87% edges could be changed dramatically and unconstrainedly.

From Figures 13, 14 and 15, it is obvious that even a large amount of targeted pairs in  $H$  which need keep exactly the same shortest paths and the close lengths of the shortest paths, 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 pairs are exactly kept after perturbation, respectively.

## 5 Conclusion and Future Plan.

In consideration of the privacy issue in social network data mining techniques, the link's weights between social network entities are sensitive in some cases such as the business transaction expenses. This paper addresses a balance between protection of sensitive weights of network links (edges) and some global structure utilities such as the short-

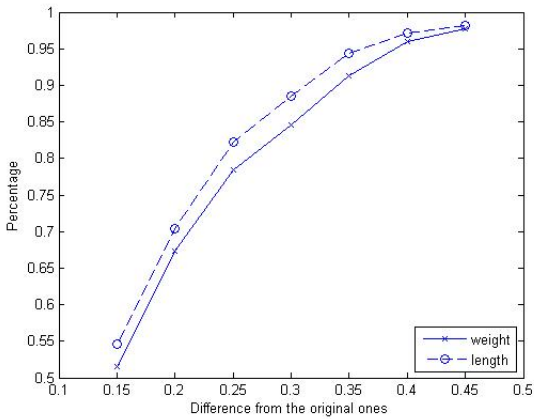


Figure 12: Percentage of the perturbed shortest path lengths and weights in the range after the Gaussian perturbation with  $\sigma=0.2$  on EIES.

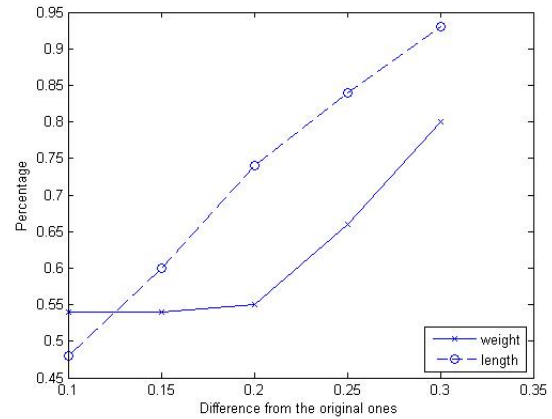
est paths and the corresponding shortest path lengths.

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.

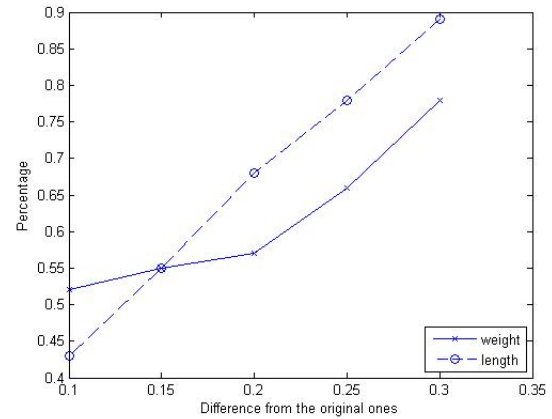
Further research work along this line can be carried out to extend our perturbation strategies to perturb weights of the original edges in case of a dynamic evolutionary complex social network in which the social network structure and its weights change over time.

## References

- [1] S. Bapna and A. Gangopadhyay, *A wavelet-based approach to preserve privacy for classification mining*, Decision Sciences Journal, 37(4):623-642, 2006.
- [2] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, 1st ed., MIT Press, 1990.
- [3] A. Evfimievski, *Randomization in privacy preserving data mining*, ACM SIGKDD Explorations Newsletter, 4(2):43-48, 2002.
- [4] K. Faust and S. Wasserman, *Social Network Analysis: Methods and Applications*, Cambridge University Press, New York, NY, 1994.
- [5] 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.
- [6] M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava, *Anonymizing social networks*, University of Massachusetts, Amherst, MA, Tech. Rep. 07-19, 2007.
- [7] A. Inkpen, *The Japanese corporate network transferred to North America: implications for North American firms*, The International Executive, 36(4): 411-433, 1994.
- [8] L. Liu, J. Wang, and J. Zhang, *Wavelet-based data perturbation for simultaneous privacy-preserving and statistics-preserving*, in Proceedings of the 2008 IEEE International Conference on Data Mining Workshops, pp. 27-35, Pisa, Italy, Dec 2008.
- [9] 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.
- [10] K. Muralidhar, R. Parsa, and R. Sarathy, *A general additive data perturbation method for database security*, Management Science, 45(10): 1399-1415, 1999.
- [11] J. M. Read and M. J. Keeling, *Disease evolution on networks:*

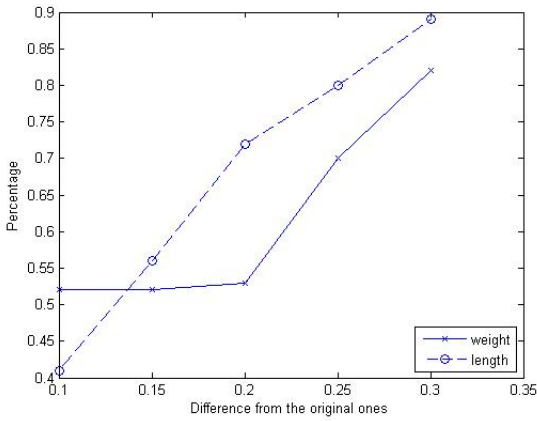


(a) EIES

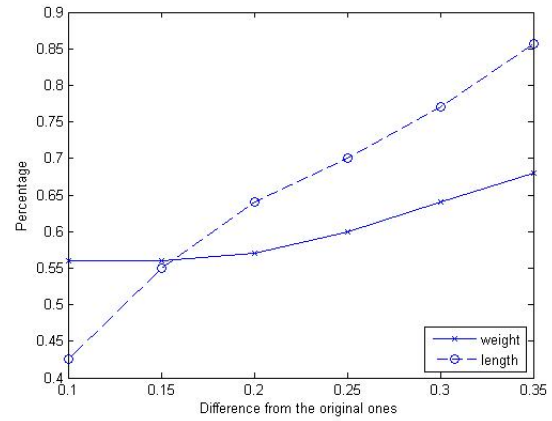


(b) Synthetics

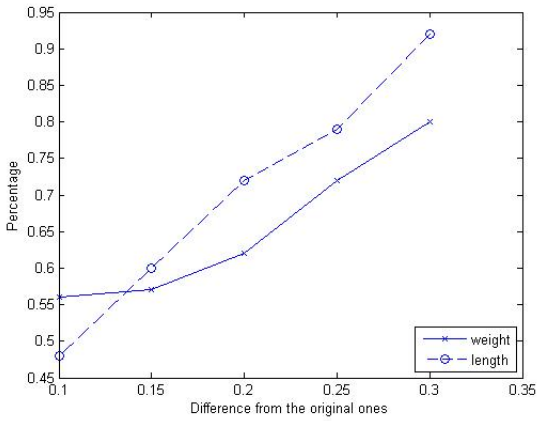
Figure 13: Percentage of the perturbed shortest path lengths and weights in the range after the greedy perturbation with 77% targeted pairs being preserved.



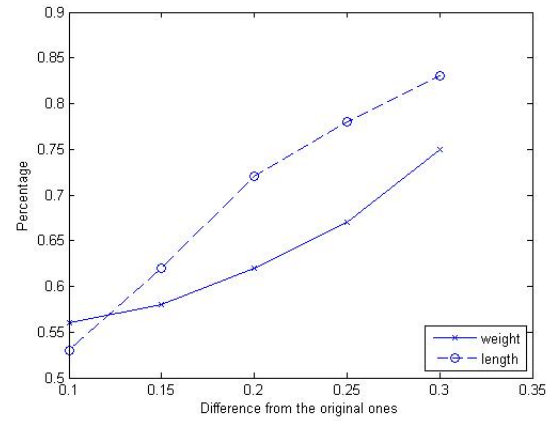
(a) EIES



(a) EIES



(b) Synthetics



(b) Synthetics

Figure 14: Percentage of the perturbed shortest path lengths and weights in the range after the greedy perturbation with 54% targeted pairs being preserved.

Figure 15: Percentage of the perturbed shortest path lengths and weights in the range after the greedy perturbation with 25% targeted pairs being preserved.

*the role of contact structure*, Proc. R. Soc. Lond. B, 270: 699-708, 2003.

- [12] E. M. Rogers, *Diffusion of Innovations*, 5th ed., Simon & Shuster, Inc., 2003.
- [13] S. M. Stigler, *Statistics on the Table*, Harvard University Press, 1999.
- [14] L. Sweeney, *Guaranteeing anonymity when sharing medical data, the DataFly system*, Journal of the American Medical Informatics Association, Suppl. S, pp. 51-55, 1997.
- [15] J. Tang, D. Zhang, and L. Yao, *Social network extraction of academic researchers*, in Proceedings of 2007 IEEE International Conference on Data Mining (ICDM 2007), Omaha, NE, Oct, 2007.
- [16] 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.
- [17] S. Xu, J. Zhang, D. Han, and J. Wang, *Singular value decom-*

*position based data distortion strategy for privacy protection*, Knowledge and Information Systems, 10(3):383-397, 2006.

- [18] 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.
- [19] E. Zheleva and L. Getoor, *Preserving the privacy of sensitive relationships in graph data*, in Proceedings of the 1st ACM SIGKDD International Workshop on Privacy, Security, and Trusting KDD, San Jose, California, pp. 153-171, Aug 2007.
- [20] 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.