2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining

Privacy Preservation by k-Anonymization of Weighted Social Networks

Maria E. Skarkala, Manolis Maragoudakis,
Stefanos Gritzalis and Lilian Mitrou
Department of Information and Communication Systems Engineering
University of the Aegean
Karlovassi, Samos 83200, Greece
Email: {mes, mmarag, sgritz, l.mitrou}@aegean.gr

Hannu Toivonen and Pirjo Moen
Department of Computer Science and HIIT
University of Helsinki, Finland
Email: {hannu.toivonen, pirjo.moen}@cs.helsinki.fi

***Abstract*—Privacy preserving analysis of a social network aims at a better understanding of the network and its behavior, while at the same time protecting the privacy of its individuals. We propose an anonymization method for weighted graphs, i.e., for social networks where the strengths of links are important. This is in contrast with many previous studies which only consider unweighted graphs. Weights can be essential for social network analysis, but they pose new challenges to privacy preserving network analysis. In this paper, we mainly consider prevention of identity disclosure, but we also touch on edge and edge weight disclosure in weighted graphs. We propose a method that provides k-anonymity of nodes against attacks where the adversary has information about the structure of the network, including its edge weights. The method is efficient, and it has been evaluated in terms of privacy and utility on real word datasets.**

I. Introduction

Social networks can be analyzed and mined for various interesting questions, such as how opinions spread or how communities evolve. As social networks are released, there is an increasing concern about the privacy of individuals involved [12], [18]. Therefore, before publishing networks, it is necessary to ensure that they do not disclose sensitive information, such as identities of the individuals or their relationships [20]. The problem is not trivial, since simply removing all identifiers or replacing them with anonymous ones is insufficient [1], [5].

Potential privacy breaches in social networks can be categorized into three types: *identity disclosure*, *link disclosure*, and *content disclosure* [3]. Individual identity is disclosed when an adversary is able to identify a target individual in the network. Link disclosure occurs when the existence of a relationship between two given individuals can be uncovered. Content disclosure, in turn, means that data associated with a node (individual) or an edge (relationship between two individuals) are made available.

In many social networks, edge weights have a special role since social ties between individuals may be stronger or weaker [8]. However, edge weights pose challenges for network anonymization. On one hand, an adversary who has information about the edge weights of individuals can use that information to attack the network. At the same time, since edge weights are crucial for many analyses of weighted graphs, as much information should be maintained as possible, to preserve the utility of the anonymized data [17].

We propose a novel k-anonymization [14] method of weighted social network data, to prevent identity disclosure by making each individual indistinguishable from at least $k-1$ other individuals. In networks, these techniques can be categorized into two general categories: (1) clustering or generalization-based and (2) graph modification-based ones. Our approach falls in the first category: it groups similar nodes (and edges) together and generalizes them to be identical and, thus, indistinguishable within each group. The proposed method is inspired by work on network compression [15]. The second approach, in turn, modifies the topological structure of the graph to make nodes structurally identical. The generalization-based approach we use gives a stronger protection against identity disclosure, as it includes degree anonymity, weight bag anonymity, as well as neighborhood anonymity as its special cases.

In addition to preventing identity disclosure in weighted networks, our method can optionally prevent edge disclosure and edge weight disclosure, if needed. For both of them, the k-anonymization process usually introduces some uncertainty. The amount of this uncertainty in an anonymized graph can be quantified, and also increased if needed to achieve the required privacy level.

The rest of the paper is structured as follows. The problem and key concepts are defined in Section II. We propose a novel algorithm for weighted graph anonymization in Section III. In Section IV we evaluate the algorithm experimentally using real data. Section V reviews previous work on privacy preservation over graphs and networks. Finally, a brief conclusion is given in Section VI.

II. Problem Definition

We model social networks as graphs that are weighted and undirected. A weighted graph is defined as a triple $G=(V,E,W)$ where V is a set of nodes, E a set of edges, and

978-0-7695-4799-2/12 $26.00 © 2012 IEEE
DOI 10.1109/ASONAM.2012.75

W a positive weight function. Given an edge $e_{i,j} = (i, j) \in E \subset V \times V$, its weight is denoted by $w_{i,j} = W(i, j)$.

We consider the following abstract problem. Given a weighted graph G, produce a privacy-preserving version G' of it.

Preventing node identity disclosure: The method we consider provides k-anonymity [14] of individuals in the network.

Definition 1: A graph is k-anonymous if every node in it is indistinguishable from at least $k - 1$ other nodes.

In our method the basic idea is that original nodes in graph G are grouped into supernodes, and edges between the original nodes are replaced by superedges between the supernodes. A supernode represents all the original nodes it contains, and a relationship between any two original nodes is described by a superedge between supernodes.

Definition 2: A supernode sn_k in an anonymized graph G' represents a set of original nodes n_i in graph G.

Definition 3: A superedge e_{sn_i,sn_j} in an anonymized graph G' represents all possible edges between nodes in supernodes sn_i and sn_j.

While a supernode is simply a group of nodes, a superedge represents a hypothetical set of edges. This set may contain edges that do not exist in the original graph G. These give rise to information loss, which we will discuss shortly.

To produce a k-anonymous generalised graph, we group the original nodes of G into supernodes of size at least k.

Definition 4: A k-anonymity grouping of a graph $G = (V, E, W)$ is a partitioning of the nodes in set V into supernodes sn_i such that $|sn_i| \geq k$.

A graph $G' = (V', E', W')$ is k-anonymous if the set V' of supernodes is a k-anonymity grouping of the set V of original nodes. This follows from the fact that a supernode represents all of its original nodes, so that the at least k original nodes within a supernode become undistinguishable.

A k-anonymous graph G' now consists of supernodes and superedges. For analysis, the process of grouping nodes and edges has to reversed to recover an approximate copy of the original graph. Original nodes are easily recovered from supernodes, but edges and their weights may have changed (we will return to this shortly). In particular, a superedge is associated with the count of true edges it represents but not their identities. To obtain an approximation of the original graph, that count of edges is materialized among all the possible edges a superedge represents. Since this results in some random changes of the network topology, it is a good practice to produce a number of alternative reconstructions, analyze all of them, and study the statistics over these graphs.

The problem can now be stated more exactly as follows.

Problem: Given a weighted graph $G = (V, E, W)$, find a k-anonymous graph $G' = (V', E', W')$ such that a given information loss measure $IL(G, G')$ is minimized.

Preventing edge disclosure: Information on relationships between individuals may also be considered sensitive. In such cases, the anonymization of the network data should also protect the information on connections between individuals, i.e., to prevent edge and edge weight disclosure. However, the k-anonymity as described above is not necessarily sufficient to prevent these two types of disclosure [11].

One possibility to prevent or at least make it more difficult for an adversary to get definite information on the existence of connections between nodes, is to avoid superedges that give absolute information about the existence of original edges. Due to k-anonymity grouping, a superedge typically also represents edges that do not exist in the original graph, and therefore, this can be done easily using superedge probabilities.

Definition 5: A superedge probability p_{sn_i,sn_j} describes how certain (or uncertain) the existence of an edge between any pair of nodes included in supernodes sn_i and sn_j is.

Such a superedge probability can be defined as the percentage of original edges that are represented by a superedge. By keeping these probabilities below a given threshold p', an adversary can only infer the existence of an edge at most with confidence p', assuming that the adversary does not have any other relevant information about the original network.

Preventing edge weight disclosure: Edge weights give descriptive information about the relationships between two nodes in a graph. Such information can be seen as sensitive, and if an adversary is aware of such weights, this information can be used to identify connected target nodes even in an anonymized graph. Thus, protection of the edge weights, i.e., avoiding edge weight disclosure, should also be ensured before publishing the network data.

In the case of k-anonymity grouping, the weights assigned to the superedges are combined from the original edge weights $w_{i,j}$, since the edges $e_{i,j}$ between the original nodes are joined to the superedges. The weights of these superedges are called superedge weights w_{sn_i,sn_j}. The combination of the original edge weights can be done in different ways. In our approach, a superedge weight is defined as the average of the original edge weights, i.e., the superedge weight is

$$W'(sn_i, sn_j) = \frac{\sum_{e \in IJ} w_e}{|IJ|}, \text{ where } IJ = E \cap (sn_i \times sn_j). \tag{1}$$

Often, these superedge weights w_{sn_i,sn_j} differ from the weights $w_{i,j}$ of the original edges. Thus, exact edge weight disclosure in those cases is prevented if the adversary has no other information. If the adversary knows all but one weight, that weight can be reverse-engineered. For more systematic and controlled protection, superedge weights can be modified by a random component to add uncertainty of their real values.

Measuring information loss: So far we have not considered at all how much information is changed or lost in the process of network anonymization. Ultimately, the amount of information lost depends on the application. Internally, the anonymization algorithm also uses an information loss function: the goal is to minimize information loss under the constraint that k-anonymity has to be reached. This loss function serves as a proxy of the real and unknown information loss functions of applications.

In our algorithm, we use as information loss function the edge weight dissimilarity between the original graph G and the generalised graph G':

$$IL(G, G') = \sum_{(i,j) \in E} |W(i,j) - W'(i,j)|^2, \tag{2}$$

where $W'(i,j)$ is the weight of the superedge that represents edge (i,j). It can be shown that the sum in Equation 2 is minimized when the weight of each superedge is the average of the original edge weights as was defined in Equation 1.

III. Algorithm Description

Based on the k-anonymity model, the main goals of our proposal are to prevent identity, edge and edge weight disclosure by an adversary which possesses knowledge about the structure of the original graph. The basic steps of our proposal for social network anonymization are the following:

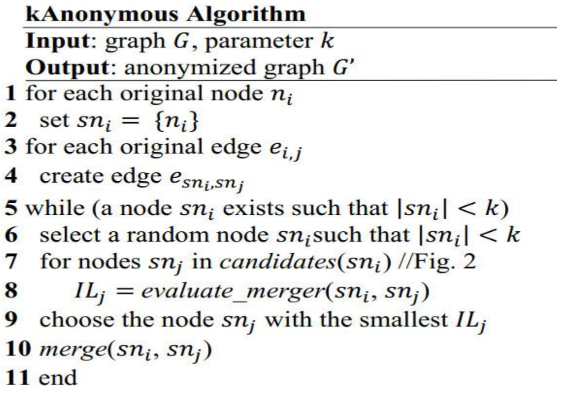
Step 1 (Naive Anonymization). All the identifiers of the original graph G are removed and replaced by temporary identities. By applying only this step, an adversary who does not possess any prior knowledge on G cannot re-identify any targeted node, edge or edge weight.

Step 2 (Node anonymity). In order to obtain a k-anonymity grouping of graph G, the original nodes of the graph are grouped to supernodes based on the similarity and strength of their relationships to other nodes. k-anonymity grouping indicates that each supernode should include at least k original nodes. The only immediate information about the supernodes is their supernode membership.

Step 3 (Edge weight computation). The original edges that connect nodes in graph G are grouped and represented by superedges between supernodes in graph G'. The weights assigned to the superedges are combined from the original edge weights $w_{i,j}$, and result from Equation 1, with the possible addition of a random component for further protection of edge weights.

Step 4 (Edge anonymity). In addition to the edge weight w', the superedges are also described by the probability of edge existence p_{sn_i,sn_j}, which defines the percentage of original edges that are represented by a superedge. If this probability is higher than p (a user given parameter) then the probability is bounded by a threshold p'.

Step 5 (Publication of the anonymized network). The final anonymized network G' is released for analysis.

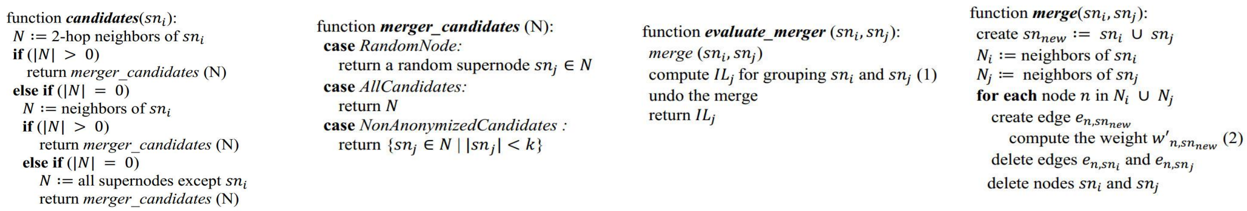


```
kAnonymous Algorithm
Input: graph $G$, parameter $k$
Output: anonymized graph $G'$
1 for each original node $n_i$
2   set $sn_i = \{n_i\}$
3 for each original edge $e_{i,j}$
4   create edge $e_{sn_i,sn_j}$
5 while (a node $sn_i$ exists such that $|sn_i| < k$)
6   select a random node $sn_i$such that $|sn_i| < k$
7   for nodes $sn_j$ in $candidates(sn_i)$ //Fig. 2
8       $IL_j = evaluate\_merger(sn_i, sn_j)$
9   choose the node $sn_j$ with the smallest $IL_j$
10 $merge(sn_i, sn_j)$
11 end
```

Figure 1: $kAnonymous$ Algorithm

The proposed method is described in $kAnonymous$ algorithm (Figure 1), which takes as input a weighted graph G and a parameter k and returns an anonymized graph G'. First, each original node n_i in graph G is described as a supernode sn_i in graph G' and the corresponding edges between the supernodes are created ($e_{sn_i,sn_j} = e_{i,j}$). Line 5 examines if there exists a supernode sn_i which contains less than k nodes, i.e. if the network stills needs to be anonymized. If yes, then on Line 6 such a supernode is picked at random. On Lines 7–10 it is merged with another supernode. To implement different strategies for selecting this other supernode, a separate function $candidates$ returns a list of possible options (see below). Each of these candidates is considered for possible merger in turn (Line 8), and finally the best candidate is chosen.

The $candidates$ function (Figure 2) returns a set of candidate nodes with which node sn_i could be merged. Since nodes with shared neighbors are likely to merge best, the set of 2-hop neighbors of node sn_i is used as candidates. In the rare occasion that this set is empty, the set of neighbors constitutes the next attempted candidate set. If this set is empty, too, then all remaining supernodes are used as candidates. The $merger\_candidates$ function further constraints the candidate set. It returns either (1) a single random supernode from the candidate set, (2) the whole set of candidates, or (3) only those candidates that are themselves not yet k-anonymous. In Section IV, we evaluate these three versions of our proposal since they give different trade-offs between speed and utility. The $evaluate\_merger$ function computes the information loss, using Equation 2, for grouping supernode sn_i with a possible candidate sn_j. Function $merge$ creates a new supernode sn_{new} as the merger of supernode sn_i with supernode sn_j, and it creates all superedges related to this new supernode and assigns them the average weight w' of the corresponding edges. The $kAnonymous$ algorithm iterates until all the supernodes in graph G' represent at least k nodes of graph G.



```
function candidates($sn_i$):
 $N$ := 2-hop neighbors of $sn_i$
 if ($|N| > 0$)
   return merger_candidates (N)
 else if ($|N| = 0$)
   $N$ := neighbors of $sn_i$
   if ($|N| > 0$)
     return merger_candidates (N)
   else if ($|N| = 0$)
     $N$ := all supernodes except $sn_i$
     return merger_candidates (N)
```

```
function merger_candidates (N):
 case RandomNode:
   return a random supernode $sn_j \in N$
 case AllCandidates:
   return $N$
 case NonAnonymizedCandidates :
   return $\{sn_j \in N \mid |sn_j| < k\}$
```

```
function evaluate_merger ($sn_i, sn_j$):
 merge ($sn_i, sn_j$)
 compute $IL_j$ for grouping $sn_i$ and $sn_j$ (1)
 undo the merge
 return $IL_j$
```

```
function merge($sn_i, sn_j$):
 create $sn_{new} := sn_i \cup sn_j$
 $N_i$ := neighbors of $sn_i$
 $N_j$ := neighbors of $sn_j$
 for each node $n$ in $N_i \cup N_j$
   create edge $e_{n,sn_{new}}$
     compute the weight $w'_{n,sn_{new}}$ (2)
   delete edges $e_{n,sn_i}$ and $e_{n,sn_j}$
  delete nodes $sn_i$ and $sn_j$
```

Figure 2: $Candidates$, $Merger\_candidates$, $Evaluate\_merger$ and $Merge$ functions

IV. Experiments

The method was evaluated using two real weighted and undirected graph datasets, the **Karate club** [19] and **Lesmis** [7]. The first dataset describes the network of 34 members of a karate club at a US university, and its 78 edges indicate social interaction among the members. The second dataset describes the network of co-appearances of 77 characters in Victor Hugo's novel "Les Miserables". The 254 edges represent the connections between any pair of characters that appear in the same chapter of the book. The weights of the edges are the number of such co appearances. For the evaluation, small datasets were used, for the reason that we wanted the first evaluation of our proposal to be more unambiguous. The usage of more complex datasets is one of our future plans.

Our aim is to preserve the utility of an anonymized graph in a high level for a better analysis. For that purpose we measure the utility in terms of general structural properties of weighted graphs, i.e. the *degree* and *volume distribution* of all nodes in the graph, the *edge weight distribution* of all edges in the graph and the *path length distribution* between all pair of nodes. The volume of a node is the sum of the weights of its adjacent edges. The path length distribution is computed for the shortest paths for all pairs of nodes. Given an anonymized graph G', we produced random instances of it according to the probabilities of edge existence associated with superedges in G'. The four statistical properties of a graph were measured for the three versions ($merger\_candidates$) of the proposed algorithm described in Section III.

The algorithm was implemented using Java. The experiments were conducted on a computer system with 1,60GHz AMD E-350 Processor and 4GB RAM running the Windows 7 operating system.

A. Running Performance

The running time of the three versions of the proposed algorithm were evaluated using different values of k for the two datasets (results not shown of the sake of brevity). As expected, the random version requires less time than the other two versions since it simply picks a candidate by random. The $NonAnonymizedCandidates$ version requires less time than the $AllCandidates$ one, especially when the k parameter is getting bigger. The k parameter has a relatively small affect in the running times.

B. Statistical properties

Figures 3 to 6 compare the utilities of weighted graphs anonymized by the three versions of our method, for the four structural measures of utility, and using different values of parameter k.

The degree distribution of both datasets resulted from the three anonymized versions and for all k parameters tend to be similar to the original. In both datasets and for all parameter values k, the edge weight distribution is maintained in a high level by both the $NonAnonymizedCandidates$ and the $AllCandidates$ versions. For both datasets, the $AllCandidates$ version performs better for $k = 5$, but on the other hand, the $NonAnonymizedCandidates$ version converse with the original path length distribution, for $k = 10$. For both datasets, versions $NonAnonymizedCandidates$ and $AllCandidates$ preserve the volume of nodes, having almost the same distribution as the original one.

While the value of k is increasing, the degree of nodes in G' is decreasing in relation to the degree of nodes in G, since the number of nodes included into the supernodes depends on the k parameter, lowering at the same time the number of a node's connections. The volume of nodes is also affected by the decrement of the degree. The edge weights are decreasing as the value of k is getting higher, due to the grouping of nodes. The path length of the original graph is preserved more for higher k.

Our results demonstrate that the *NonAnonymizedCandidates* version preserves privacy, and accurate results can be exported from the analysis of the anonymized graph. Both the *AllCandidates* and *NonAnonymizedCandidates* versions can preserve three out of four statistical properties in the same way.

V. Related work

Campan and Truta [2] use an approach similar to ours, but they consider unweighted graphs. They also consider node attributes; this method could be combined with ours. In the same spirit, Hay et al. [6] present a k-anonymous edge generalization approach for unweighted graphs.

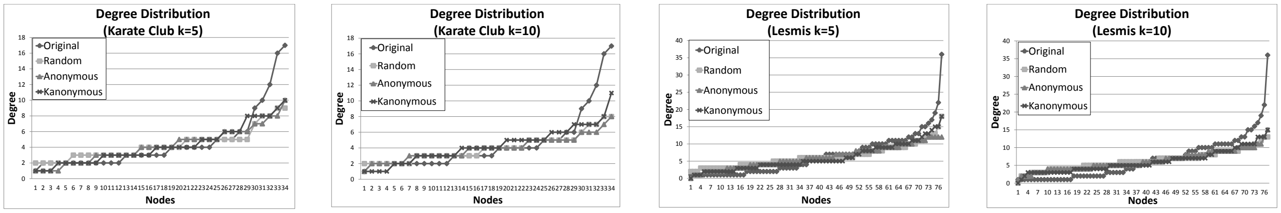


Figure 3: Degree Distribution

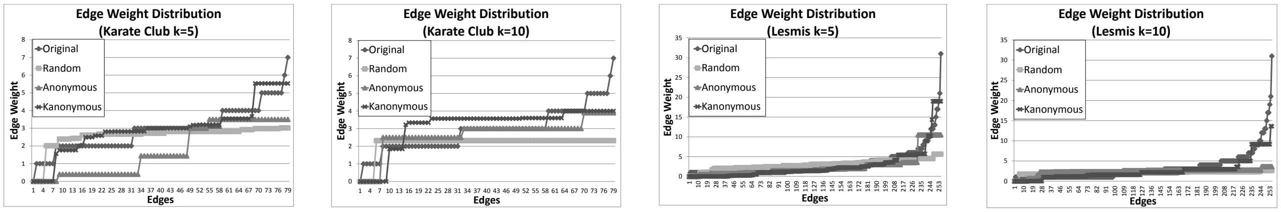


Figure 4: Edge Weight Distribution

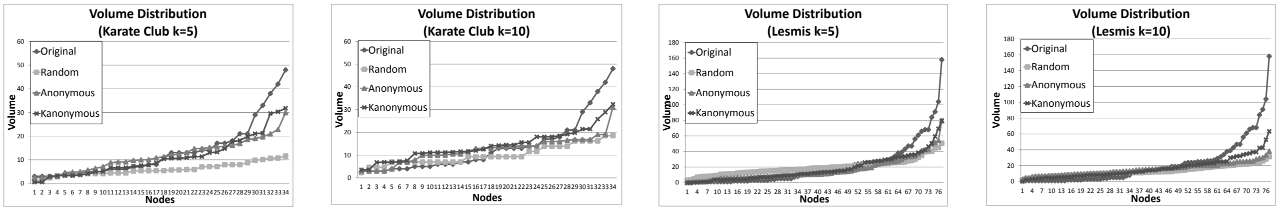


Figure 5: Volume Distribution

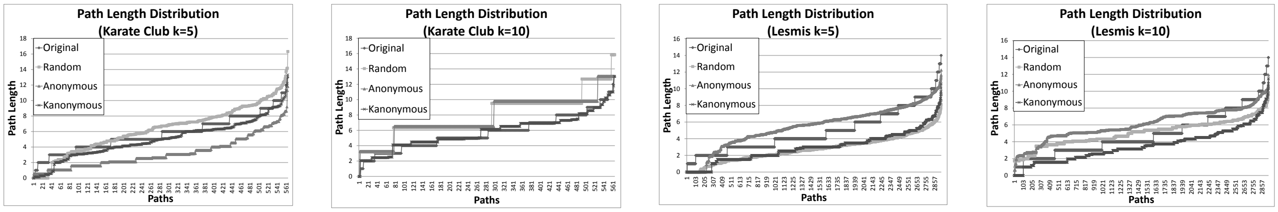


Figure 6: Path Length Distribution

Most of the existing literature on network anonymization deals with privacy preservation on simple undirected and unweighted graphs. There are, however, some methods for weighted networks.

Liu et al. (e.g., [13]) have studied anonymization techniques for weighted graphs to preserve linear properties such as shortest paths between pairs of nodes. Das et al. [4] propose a linear programming method to anonymize the weights of edges while preserving shortest paths. Li et al. [8] propose volume and histogram anonymization to prevent weight-based attacks by modifying edges and edge weights. They also propose a volume sequence perturbation model for weight anonymization [9]. A greedy algorithm is proposed by Wang et al. [16] to preserve sensitive paths. The method perturbs a minimal number of edge weights so that there are at least k indistinguishable shortest paths.

A generalization-based approach that preserves a network from identity and edge weight disclosure, is presented by Liu et al. [10]. Their method groups together nodes with similar weight bags, i.e., with a similar set of weights, not with a similar set of neighbors. For a fixed k, our model gives less information to the adversary, but may also have lower utility.

Comparing these approaches is a topic for future work.

VI. Conclusion

Publishing social network data for analysis by researchers while preserving the privacy of the individuals involved has raised many concerns. In this paper we presented a clustering-based k-anonymization technique for weighted network data. The method groups nodes with similar sets of neighbors and their connections into supernodes and superedges, respectively. At the same time, the method tries to preserve utility of the graph. Experimental results suggest that the approach can find a balance between privacy and utility for real world weighted graphs. Further research work can be carried out to investigate the performance of our proposal in more complex social networks, its effectiveness on other statistical graph properties, and to compare it to other approaches.

Acknowledgements

This work has been supported by the Algorithmic Data Analysis (Algodan) Center of Excellence of the Academy of Finland.

References

[1] Backstrom L., Dwork C., and Kleinberg J. "Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography". In Proceedings of the 16th international conference on World Wide Web (WWW '07). ACM, New York, NY, USA, pp. 181-190, 2007.

[2] Campan A. and Truta T.M. "A clustering approach for data and structural anonymity in social networks". In Proceedings of the 2nd ACM SIGKDD International Workshop on Privacy, Security, and Trust in KDD (PinKDD'08), in Conjunction with KDD'08, Las Vegas, Nevada, USA, pp. 1-10, 2008.

[3] Clarkson K., Liu K., and Terzi E. "Towards identity anonymization in social networks". In Link Mining: Models Algorithms and Applications, Chapter 14, pp. 359 385, Springer, 2010.

[4] Das, S., Omer E. and Abbadi, A. E. "Anonymizing edge-weighted social network graphs". Technical report, Department of Computer Science, University of California Santa Barbara, USA, 2009.

[5] Hay M., Miklau G., Jensen D., Weis P., and Srivastava S. "Anonymizing social networks". Technical Report 07-19, University of Massachusetts Amherst, 2007.

[6] Hay M., Miklau G., Jensen D., Towsley D., and Weis P. "Resisting structural re-identification in anonymized social networks". In Proc. VLDB Endow. vol. 1, no. 1, pp. 102-114, 2008.

[7] Knuth D.E. "The Stanford GraphBase: A platform for Combinatorial Computing", Addison-Wesley, Reading, MA, 1993.

[8] Li Y. and Shen H. "Anonymizing Graphs Against Weight-Based Attacks". In Proceedings of the 2010 IEEE International Conference on Data Mining Workshops (ICDMW '10). IEEE Computer Society, Washington, DC, USA, pp. 491-498, 2010.

[9] Li Y. and Shen H. "On Identity Disclosure in Weighted Graphs". In Proceedings of the 2010 International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT '10). IEEE Computer Society, Washington, DC, USA, pp. 166-174, 2010.

[10] Liu X. and Yang X. "A generalization based approach for anonymizing weighted social network graphs". In Proceedings of the 12th international conference on Web-age information management (WAIM'11), Springer-Verlag Berlin, Heidelberg, pp. 118-130, 2011.

[11] Liu, L., Liu, J., and Zhang, J. "Privacy preservation of affinities in social networks". In Proceedings of the International conference on Information Systems, pp. 372-376, Porto, Portugal, 2010.

[12] Liu K., Das K., Grandison T., and Kargupta H. "Privacy-Preserving Data Analysis on Graphs and Social Networks". In Next Generation of Data Mining, Chapter 21, pp. 419-437, Chapman & Hall/CRC, December 2008.

[13] Liu, L., Wang, J., Liu, J., and Zhang, J. "Privacy preservation in social networks with sensitive edge weights". In SIAM Data Mining Conference (SDM 2009), pp. 954965, 2009.

[14] Samarati P., "Protecting Respondent's Privacy in Microdata Release", IEEE Transactions on Knowledge and Data Engineering, vol. 13, n. 6, pp. 1010-1027, November 2001.

[15] Toivonen H., Zhou F., Hartikainen A., and Hinkka A. "Compression of weighted graphs". In Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '11). ACM, New York, NY, USA, pp. 965-973, 2011.

[16] Wang S.L., Tsai Z.Z., Hong T.P., and Ting I.H. "Anonymizing shortest paths on social network graphs". In Proceedings of the Third international conference on Intelligent information and database systems - Volume Part I (ACIIDS'11), Springer-Verlag Berlin, Heidelberg, pp. 129-136, 2011.

[17] Watanabe C., Amasaga T. and Liu L.. "Privacy Risks and Countermeasures in Publishing and Mining Social Network Data". In Proceedings of International Conference on Collaborative Computing (CollaborateCom 2011), Orlando, FL.USA, pp. 55-66, Oct. 2011.

[18] Wu X., Ying X., Liu K., and Chen L. "A survey of algorithms for privacy- preservation of graphs and social networks" In Managing and Mining Graph Data. Chapter 14, Kluwer Academic Publishers, vol. 40, pp. 421-453, 2010

[19] Zachary W. "An information flow model for conflict and fission in small groups". Journal of Anthropological Research, vol. 33, no. 4, pp. 452-473, 1977.

[20] Zou L., Chen L., and Ozsu M.T. "k-automorphism: a general framework for privacy preserving network publication". Proc. VLDB Endow. 2, 1, pp. 946-957, 2009.