A Summary of k-Degree Anonymous Methods for Privacy-Preserving on Networks

Jordi Casas-Roma[1], Jordi Herrera-Joancomartí[2], and Vicenç Torra[3]

[1] Universitat Oberta de Catalunya
Barcelona, Spain
jcasasr@uoc.edu
[2] Universitat Autònoma de Barcelona
Bellaterra, Spain
jherrera@deic.uab.cat
[3] Artificial Intelligence Research Institute (IIIA)
Spanish National Research Council (CSIC)
Bellaterra, Spain
vtorra@iiia.csic.es

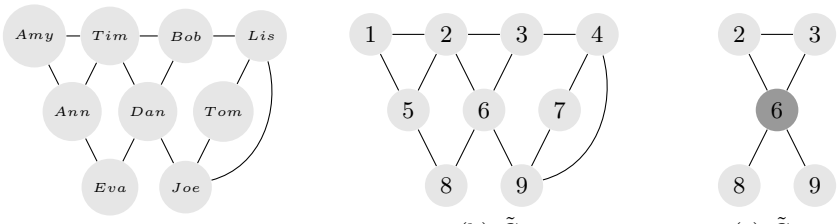
Abstract. In recent years there has been a significant raise in the use of graph-formatted data. For instance, social and healthcare networks present relationships among users, revealing interesting and useful information for researches and other third-parties. Notice that when someone wants to publicly release this information it is necessary to preserve the privacy of users who appear in these networks. Therefore, it is essential to implement an anonymization process in the data in order to preserve users' privacy. Anonymization of graph-based data is a problem which has been widely studied last years and several anonymization methods have been developed. In this chapter we summarize some methods for privacy-preserving on networks, focusing on methods based on the k-anonymity model. We also compare the results of some k-degree anonymous methods on our experimental set up, by evaluating the data utility and the information loss on real networks.

Keywords: Privacy, K-Anonymity, Social networks, Graphs, Information loss, Data utility

1 Introduction

Nowadays, large amounts of data are being collected on social and other kinds of networks, which often contain personal and private information of users and individuals. Although basic processes are performed on data anonymization, such as removing names or other key identifiers, the remaining information can still be sensitive and useful for an attacker to re-identify users and individuals. To solve this problem, methods which introduce noise to the original data have been developed in order to hinder the subsequent processes of re-identification. A natural strategy for protecting sensitive information is to replace identifying attributes with synthetic identifiers. We refer to this procedure as simple or naïve

anonymization. This common practice attempts to protect sensitive information by breaking the association between the real-world identity and the sensitive data.



(a) G (b) $\tilde{G}$ (c) $\tilde{G}_{Dan}$

Fig. 1: Naïve anonymization of a toy network, where G is the original graph, $\tilde{G}$ is the naïve anonymous version and $\tilde{G}_{Dan}$ is Dan's 1-neighbourhood.

Figure 1a shows a toy example of a social network, where each vertex represents an individual and each edge indicates the friendship relation between them. Figure 1b presents the same graph after a naïve anonymization process, where vertex identifiers have been removed and the graph structure remains the same. One can think users' privacy is secure, but an attacker can break the privacy and re-identify a user on the anonymous graph. For instance, if an attacker knows that Dan has four friends and two of them are friends themselves, then he can construct the 1-neighbourhood of Dan, depicted in Figure 1c. From this sub-graph, the attacker can uniquely re-identify user Dan on anonymous graph. Consequently, user's privacy has been broken by the attacker.

Two types of attacks have been proposed which show that identity disclosure would occur when it is possible to identify a sub-graph in the released graph in which all the vertex identities are known [2]. In the active attack an adversary creates k accounts and links them randomly, then he creates a particular pattern of links to a set of m other users that he is interested to monitor. The goal is to learn whether two of the monitored vertices have links between them. When the data is released, the adversary can efficiently identify the sub-graph of vertices corresponding to his k accounts with high probability. With as few a $k = \mathcal{O}(log(n))$ accounts, an adversary can recover the links between as many as $m = \mathcal{O}(log^2(n))$ vertices in an arbitrary graph of size n. The passive attack works in a similar manner. It assumes that the exact time point of the released data snapshot is known, and that there are k colluding users who have a record of what their links were at that time point. Other attacks on naively anonymized network data have been developed, which can re-identify vertices, disclose edges between vertices, or expose properties of vertices (e.g., vertex features). These attacks include: matching attacks, which use external knowledge of vertex features [26] [41] [39]; injection attacks, which alter the network prior to publication [2]; and auxiliary network attacks, which use publicly available networks as an external

information source [28]. To solve these problems, methods which introduce noise to the original data have been developed in order to hinder the subsequent processes of re-identification.

In this chapter we will summarize some methods for privacy-preserving on networks, specifically, we will focus on methods based on the concept of k-anonymity model. This model is widely used for data privacy, both for relational and graph-formatted data. We will also compare four k-anonymous methods in terms of data utility and information loss on undirected and unlabelled real networks.

This chapter is organized as follows. In Section 2, we review the state of the art of anonymization on networks, specifically the k-degree anonymous methods. Section 3 introduces the four tested algorithms for k-degree anonymity on networks. Then, in Section 4, we compare our tested algorithms among them, in terms of information loss and data utility, and discuss the results. Lastly, in Section 5, we present the conclusions.

1.1 Notation

Let $G = (V, E)$ be a simple, undirected and unlabelled graph, where V is the set of vertices and E the set of edges in G. We define $n = |V|$ to denote the number of vertices and $m = |E|$ to denote the number of edges. We use d to define the degree sequence of G, where d is a vector of length n and d_i is the value of i-th element, that is, the degree of vertex $v_i \in V$. We refer to the ordered degree sequence as a monotonic non-decreasing sequence of the vertex degrees, that is $d_i \leq d_j \; \forall i < j$. We denote the set of 1-neighbourhood of vertex v_i as $\Gamma(v_i)$, i.e, $\Gamma(v_i) = \{v_j : (v_i, v_j) \in E\}$. Finally, we designate $G = (V, E)$ and $\tilde{G} = (\tilde{V}, \tilde{E})$ to refer the original and the anonymous graphs, respectively.

2 Privacy-preserving on networks

Zhou and Pei [39] noticed that to define the problem of privacy preservation in publishing social network data, we need to formulate the following issues: Firstly, we need to identify the privacy information to be preserved. Secondly, we need to model the background knowledge that an adversary may use to attack the privacy. Thirdly, we need to specify the usage of the published social network data so that an anonymization method can try to retain the utility as much as possible while the privacy information is fully preserved.

Regarding to the privacy information to be preserved, we point out three main categories of privacy breaches in social networks:

1. *Identity disclosure* occurs when the identity of an individual who is associated with a vertex is revealed.
2. *Link disclosure* occurs when the sensitive relationship between two individuals is disclosed.

3. *Attribute disclosure* which seeks not necessarily to identify a vertex, but to reveal sensitive labels of the vertex. The sensitive data associated with each vertex is compromised.

Identity disclosure and link disclosure apply on all types of networks. However, attribute disclosure only applies on edge-labelled networks. In addition, link disclosure can be considered a special type of attribute disclosure, since edges can be seen as a vertex attributes. In this text, we will focus on identity disclosure.

From a high level view, there are three general families of methods for achieving network data privacy. The first family encompasses "graph modification" methods. These methods first transform the data by edges or vertices modifications (adding and/or deleting) and then release them. The data is thus made available for unconstrained analysis. The second family encompasses "generalization" or "clustering-based" approaches. These methods can be essentially regarded as grouping vertices and edges into partitions called super-vertices and super-edges. The details about individuals can be hidden properly, but the graph may be shrunk considerably after anonymization, which may not be desirable for analysing local structures. The generalized graph, which contains the link structures among partitions as well as the aggregate description of each partition, can still be used to study macro-properties of the original graph. Among others, [20] [5] [29] [14] and [3] are interesting approaches to generalization concept. Finally, the third family encompasses "privacy-aware computation" methods, which do not release data, but only the output of an analysis computation. The released output is such that it is very difficult to infer from it any information about an individual input datum. For instance, differential privacy [16] is a well-known privacy-aware computation approach. Differential private methods refer to algorithms which guarantee that individuals are protected under the definition of differential privacy, which imposes a guarantee on the data release mechanism rather than on the data itself. The goal is to provide statistical information about the data while preserving the privacy of users. Interesting works can be found, among others, in [22], [21] and [15].

2.1 Graph modification approaches

Graph modification approaches anonymize a graph by modifying (adding and/or deleting) edges or vertices in a graph. These modifications can be made randomly or in order to fulfil some desired constraints. The first methods are called randomization methods and are based on adding random noise in the original data. They have been well investigated for relational data. Naturally, edge randomization can also be considered as an additive-noise perturbation. Notice that the randomization approaches protect against re-identification in a probabilistic manner. Hay et al. [19] proposed a method to anonymize unlabelled graphs based on randomly removing m edges and then randomly adding m fake edges. Ying and Wu [36] propounded two algorithms specifically designed to preserve spectral characteristics of the original graph. Ying et al. [35] presented a method

which divides the graph into blocks according to the degree sequence and implements modifications (by adding and removing edges) on the vertices at high risk of re-identification, not at random over the entire set of vertices. Boldi et al. [4] introduced a new anonymization approach that is based on injecting uncertainty in social graphs (they add or remove edges partially with a certain probability) and publishing the resulting uncertain graphs. Other approaches consider the degree sequence of the vertices or other structural graph characteristics (for example, transitivity or average distance between pairs of vertices) as important features which the anonymization process has to keep as equal as possible on anonymized network [17] [37].

2.2 *k*-anonymity model

Other ways to anonymize consider graph modification methods to meet desired privacy constraints. The notion of k-anonymity [32] [30] is included in this group, though it was introduced for the privacy preservation on relational data. Formally, the k-anonymity model is defined as follows. Let $RT(A_1, \ldots, A_n)$ be a table and QI_{RT} be the quasi-identifier associated with it. RT is said to satisfy k-anonymity if and only if each sequence of values in $RT\,[QI_{RT}]$ appears with at least k occurrences in $RT\,[QI_{RT}]$. The k-anonymity model indicates that an attacker cannot distinguish between different k records although he manages to find a group of quasi-identifiers. Therefore, the attacker cannot re-identify an individual with a probability greater than $\frac{1}{k}$. In general, the higher the k value, the greater the anonymization and also the information loss. Ying et al. [35] demonstrated that deliberate k-anonymization can preserve structural properties of networks much better than the randomization techniques.

The k-anonymity model can be applied using different quasi-identifiers when dealing with networks rather than relational data. A widely used option is to consider the vertex degree as a quasi-identifier, i.e, this model presumes that the only possible attack is when the attacker knows the degree of some target vertices. This corresponds to k-degree anonymity. Therefore, if some vertices are re-identified using this information, then we have an information leakage. Liu and Terzi [26] developed a method to create a k-degree anonymous network $\tilde{G} = (V, \tilde{E})$ from the original network $G = (V, E)$ and an integer k, where $\tilde{E} \cap E \approx E$. Their method is based on anonymizing the degree sequence by linear programming techniques. Casas-Roma et al. [8] presented a method based on evolutionary algorithms, which anonymizes the degree sequence and then translates the modifications to the edge set. Chester et al. [10] [12] also considered the k-degree anonymity problem, but they modified the network structure by adding new edges between fake and real vertices or between fakes vertices. Under the constraint of minimum vertex additions, they show that on vertex-labelled networks, the problem is NP-complete. Casas-Roma et al. [6] introduced an algorithm specifically designed for k-degree anonymity on large networks. The authors construct a k-degree anonymous network by the minimum number of edge modifications using univariate micro-aggregation to anonymize the degree

sequence, and then they modify the graph structure using basic operations for graph modification to meet the k-degree anonymous sequence.

Chester et al. [11] introduced the concept of k-subset-degree anonymity as a generalization of the notion of k-degree-anonimity. In k-subset-anonymity problem the goal is to anonymize a given subset of vertices, while adding the fewest possible number of edges. Formally, k-degree-subset-anonymity problem is defined as given an input graph $G = (V, E)$ and an anonymous subset $X \subseteq V$, produces an output graph $\tilde{G} = (V, E \cup \tilde{E})$ such that X is k-degree-anonymous and $|\tilde{E}|$ is minimized. They presented an algorithm to k-subset-degree-anonymity which is based on using the degree constrained sub-graph satisfaction problem. For unlabelled networks, they give a near-linear algorithm ($\mathcal{O}(nk)$). The output of the algorithm is an anonymized version of G where enough edges have been added to ensure all the vertices in X have the same degree as at least $k-1$ others.

Zhou and Pei [39] [40] introduced the 1-neighbourhood sub-graph of the objective vertices as a quasi-identifier. For a vertex $u \in V$, u is k-anonymous in G if there are at least $k-1$ other vertices $v_1, \ldots, v_{k-1} \in V$ such that $\Gamma(u), \Gamma(v_1), \ldots, \Gamma(v_{k-1})$ are isomorphic. G is k-anonymous if every vertex is k-anonymous in G. It is called k-neighbourhood anonymity. Tripathy and Panda [33] noted that their algorithm cannot handle the situations in which an adversary has knowledge about vertices in the second or higher hops of a vertex, in addition to its immediate neighbours. To handle this problem, they proposed a modification to their algorithm to handle such situations. In addition, the time complexity of their algorithm is less than that of Zhou and Pei. Zou et al. [41] considered all structural information about a target vertex and propounded a new model called k-automorphism. Hay et al. [20] go a step further and proposed a method, named k-candidate anonymity, that uses queries as quasi-identifier. In this method, a vertex v_i is k-candidate anonymous to question Q if there are at least $k-1$ others vertices in the network with the same answer. Cheng et al. [9], in their work on k-isomorphism, formed k pairwise isomorphic sub-graphs to achieve protection against two specific classes of attacks. Wu et al. [34] introduced the k-symmetry model, wherein for any vertex v, there exists at least $k-1$ other vertices to which v can be mapped using an automorphism of the underlying graph. Kapron et al. [23] analysed the problem of anonymizing an edge-labelled network. They considered the label sequence $S_v = (\ell_1, \ell_2, \ldots, \ell_m)$ of a vertex v as some ordering of the labels of the edges incident on v. Lastly, Stokes and Torra [31] introduced the concept of n-confusion as a generalization of k-anonymity and a new definition of (k,ℓ)-anonymous graph, which they proved to have severe weaknesses. The authors also presented a set of algorithms for k-anonymization of graphs.

When there is little diversity in the sensitive attributes inside an equivalence class, it is possible to obtain information from anonymized data. Although there are k indistinguishable records in each equivalence class, if the information in sensitive attributes is the same, then it is possible to infer information unless the attacker does not know exactly which record it is. The ℓ-diversity model

[27] alleviates the problem of sensitive attribute disclosure. It ensures that the sensitive attribute value in each equivalence class are diverse. But an attacker can also infer some sensible information from similarity or skewness attack [25]. This leads to t-closeness [25], which is another privacy definition that considers the sensitive attribute distribution in each class. There are other privacy definitions of this flavour, but they are all been criticized for being ad hoc [38].

Chester et al. [13] study the complexity of anonymization on different kinds of network (labelled, unlabelled and bipartite). For general, edge-labelled graphs, label sequence subset anonymization (and thus table graph anonymization, k-neighbourhood anonymity, i-hop anonymity and k-symmetry) are NP-complete for $k \geq 3$. For bipartite, edge-labelled graphs, label sequence subset anonymization is in P for $k = 2$ and is NP-complete for $k \geq 3$. For bipartite, unlabelled graphs, degree-based subset anonymization is in P for all values of k. And for general, vertex-labelled graphs, they show that vertex label sequence-based anonymization, and consequently t-closeness, is NP-complete.

3 k-Degree anonymous methods

We have selected four relevant methods for k-degree anonymity on networks. In subsequent sections, we will analyse these methods and compare the empirical results on real networks. Firstly, Liu and Terzi defined the concept of k-degree anonymity and presented their method in [26]. Secondly, Casas-Roma et al. introduced two algorithms, the first one based on evolutionary algorithms [8] and the second one based on univariate micro-aggregation [6]. Lastly, Chester et al. propounded an algorithm based on vertex and edge addition [12]. All methods achieve the same privacy level, since they presuppose the same adversary knowledge and apply the same concept to preserve the network's privacy. Therefore, the evaluation of the results is interesting to compare the data utility and information loss on anonymous datasets.

3.1 Preliminaries

The degree sequence is an interesting tool since the concept of k-degree anonymity for a network can be directly mapped to its degree sequence, as Liu and Terzi showed in [26] and we recall in the following definitions:

Definition 1. *A vector of integers V is k-anonymous if every distinct value $v_i \in V$ appears at least k times.*

Definition 2. *A network $G = (V, E)$ is k-degree anonymous if the degree sequence of G is k-anonymous.*

Accordingly to Definition 2, the degree sequence is a key point when dealing with k-degree anonymity on networks. Regarding to the degree sequence, notice that:

- The number of elements is n, which represents the number of vertices.

– Each $d_i \in d$ must be an integer in the range $[0, n-1]$, since each d_i is the degree of vertex v_i.
– $\sum_{i=1}^{n} d_i = 2m$, since each edge is counted twice in the degree sequence. Therefore, $\sum_{i=1}^{n} d_i = \sum_{i=1}^{n} \tilde{d}_i$ if we want to keep the same number of edges in the anonymous graph.

The construction of the k-anonymous degree sequence determines the privacy level. Moreover, the distance between the original and the anonymous degree sequences is critical in terms of data utility and information loss. An optimal sequence has to provide the requested k-anonymity level and also has to minimize the distance from the original degree sequence. Some of our tested methods use Equation 1 to compute the distance between the original degree sequence and the anonymous one.

$$\Delta = \sum_{i=1}^{n} |\tilde{d}_i - d_i| \tag{1}$$

3.2 A dynamic programming algorithm

Liu and Terzi [26] developed a method based on adding and removing edges from the original graph $G = (V, E)$ in order to construct a new graph $\tilde{G} = (\tilde{V}, \tilde{E})$, which fulfil k-degree anonymity model and the vertex set remains the same, i.e, $V = \tilde{V}$. Their approach is two-step based: in the first step the degree anonymization problem is considered, and in the second step the graph construction problem is dealt.

Degree anonymization. Given the degree sequence d of the original input graph $G = (V, E)$, the algorithm outputs a k-anonymous degree sequence $(\tilde{d})$ such that the degree-anonymization cost Δ computed by Equation 1 is minimized. The authors proposed three approximation techniques to solve the degree anonymization problem. They first gave a dynamic-programming algorithm (DP) that solves the degree anonymization problem optimally in time $\mathcal{O}(n^2)$. Then, they showed how to modify it to achieve linear-time complexity. Finally, they also gave a greedy algorithm that runs in time $\mathcal{O}(nk)$.

Graph construction. The authors presented two approaches to resolve the graph construction problem. The first approach considers the following problem: Given the original graph $G = (V, E)$ and the desired k-anonymous degree sequence $\tilde{d}$ (output by the the previous step), they construct a k-degree anonymous graph $\tilde{G} = (V, \tilde{E})$ with $\tilde{E} \cap E = E$ and degree sequence equal to $\tilde{d}$. Notice that the problem definition implies that only edge addition operations are allowed. The algorithm for solving this problem was called *SuperGraph*. It takes as inputs the original graph G and the desired k-degree anonymous sequence $\tilde{d}$, operates on the sequence of additional degrees $\tilde{d} - d$ and outputs a super-graph of the original graph, if such graph exists.

The requirement that $\tilde{E} \cap E = E$ may be too strict to satisfy. Thus, the second approach considers a relaxed requirement where $\tilde{E} \cap E \approx E$, which means that most of the edges of the original graph appear in the degree-anonymous graph as well, but not necessarily all of them. The authors called this version of the problem the "Relaxed Graph Construction" problem. The *ConstructGraph* algorithm with input $\tilde{d}$, outputs a simple graph $\tilde{G}_0 = (V, \tilde{E}_0)$ with degree sequence exactly $\tilde{d}$, if such graph exists. Although $\tilde{G}_0$ is k-degree anonymous, its structure may be quite different from the original graph $G = (V, E)$. The *GreedySwap* algorithm inputs $\tilde{G}_0$ and G, and transforms $\tilde{G}_0$ into $\tilde{G} = (V, \tilde{E})$ with degree sequence equal to $\tilde{d}$ and $\tilde{E} \cap E \approx E$ using greedy heuristic techniques. At each step i, the graph $\tilde{G}_{i-1} = (V, \tilde{E}_{i-1})$ is transformed into $\tilde{G}_i = (V, \tilde{E}_i)$ such that the degree sequences are equal and $|\tilde{E}_i \cap E| > |\tilde{E}_{i-1} \cap E|$. The transformation is made using *valid swap* operations, which are defined by four vertices v_i, v_j, v_k and v_l of $\tilde{G}_i = (V, \tilde{E}_i)$ such that (v_i, v_k) and $(v_j, v_l) \in \tilde{E}_i$, and (v_i, v_j) and $(v_k, v_l) \notin \tilde{E}_i$ or (v_i, v_l) and $(v_j, v_k) \notin \tilde{E}_i$. A valid swap operation transforms $\tilde{G}_i$ to $\tilde{G}_{i+1}$ by updating the edges $\tilde{E}_{i+1} \leftarrow \tilde{E}_i \setminus \{(v_i, v_k), (v_j, v_l)\} \cup \{(v_i, v_j), (v_k, v_l)\}$ or $\tilde{E}_{i+1} \leftarrow \tilde{E}_i \setminus \{(v_i, v_k), (v_j, v_l)\} \cup \{(v_i, v_l), (v_j, v_k)\}$, as we depict in Figure 2.

Fig. 2: Valid swap operation among vertices v_i, v_j, v_k and v_l. Dashed lines represent deleted edges while solid lines are the added ones.

3.3 An univariate micro-aggregation approach

Univariate Micro-aggregation for Graph Anonymization[4] (in short, UMGA) algorithm was proposed in [6] and it was designed to achieve k-degree anonymity on large networks. The algorithm performs modifications to the original network only on edge set (E). Hence, the vertex set (V) does not change during anonymization process. In a similar way to the previous method, it is based on a two-step approach.

Degree sequence anonymization. It constructs a k-degree anonymous sequence $\tilde{d} = \{\tilde{d}_1, \ldots, \tilde{d}_n\}$ from the degree sequence $d = \{d_1, \ldots, d_n\}$ of the original network $G = (V, E)$ using Definition 1. This method uses the optimal univariate micro-aggregation [18] to achieve the best group distribution and then it

[4] Source code available at: http://deic.uab.cat/~jcasas/

computes the value for each group that minimizes the distance Δ computed by Equation 1 from the original degree sequence.

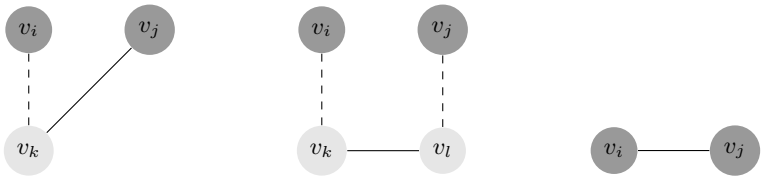
Without loss of generality, the authors assume d to be an ordered degree sequence of the original network. Otherwise, they apply a permutation f to the sequence to reorder the elements. Let k be an integer such that $1 \leq k < n$ which is the k-degree anonymity value. In order to apply the optimal univariate micro-aggregation, and according to Hansen and Mukherjee [18], the authors construct a new directed network $H_{k,n}$ and get the optimal partition which is exactly the set of groups that corresponds to the arcs of the shortest path from vertex 0 to vertex n on this network. They denote by g the optimal partition, where g has $\frac{n}{k} \leq p \leq \frac{n}{2k-1}$ groups and each of them (g_j) has between k and $2k-1$ items. Obviously, each $d_i \in d$ belongs to a specific group g_j.

Next, the algorithm computes the specific value for each group g_j, since the mean value of all group members $d_i \in g_j$ can be a real number and an integer number is needed in the degree sequence. Using the floor or ceiling functions to round these values, the total number of edges in each group g_j (computed as the sum of all $d_i \in g_j$) can be the same, higher (which means some new edges are needed) or smaller (which means some edges have to be deleted). To optimally resolve this operation two methods are proposed to achieve the best combination on reasonable time: firstly, the exhaustive method explores all possible combinations until it finds an optimal solution. Secondly, the greedy method uses a probability distribution to find a quasi-optimal (in many cases, the optimal) solution in a faster way.

Graph modification. It builds a new network $\tilde{G} = (V, \tilde{E})$ where its degree sequence is equal to $\tilde{d}$ by using basic edge modification operations. These operations allow it to modify the network's structure according to the anonymized degree sequence $(\tilde{d})$. By Definition 2 the anonymized network $\tilde{G}$ will be k-degree anonymous.

In order to modify the edge set of a given network, the authors define three basic operations: edge switch, edge removal and edge addition. The *edge switch* between three vertices can be defined as follows: if $v_i, v_j, v_k \in V$, $(v_i, v_k) \in E$ and $(v_j, v_k) \notin E$, we can delete (v_i, v_k) and create (v_j, v_k), as shown in Figure 3a. The *edge removal* is defined as follows: we select four vertices $v_i, v_j, v_k, v_l \in V$ where $(v_i, v_k) \in E$, $(v_j, v_l) \in E$ and $(v_k, v_l) \notin E$. We delete edges (v_i, v_k) and (v_j, v_l), and create a new edge (v_k, v_l), as depicted on Figure 3b. Finally, the *edge addition* is defined as follows: we select two vertices $v_i, v_j \in V$ where $(v_i, v_j) \notin E$ and create it. It is presented in Figure 3c.

The selection of the auxiliary edges is an important feature, since adding or removing important edges is critical for network structure and information flow. For instance, adding or removing a bridge-like edge may considerably reduce or increase the average distance and the shortest paths of the entire network. Two approaches were presented to select the auxiliary edges needed for graph modification process: the first one is based on random edge selection, which is the fastest way to select the auxiliary edges. The second approach is based on



(a) Edge switch (b) Edge removal (c) Edge addition

Fig. 3: Basic operations for network modification with vertex invariability. Dashed lines represent deleted edges while solid lines are the added ones.

selecting auxiliary edges by considering the relevance of each edge according to edge neighbourhood centrality (NC) [7], which identifies the most important edges on a network with low complexity ($\mathcal{O}(m)$). Obviously, this approach leads the process to a low information loss results.

3.4 Vertex addition method

Chester et al. [12, 10] focused on creating a k-degree-anonymous graph $\tilde{G} = (V \cup \tilde{V}, E \cup \tilde{E})$ from the original one $G = (V, E)$. In $\tilde{G}$, the authors require that all the original vertices (V) are k-degree-anonymous. They also require that the new vertices are concealed as well so that they cannot be readily identified and removed from the graph in order to recover G, i.e, $V \cup \tilde{V}$ is k-degree-anonymous in $\tilde{G}$. They seek to minimise $|\tilde{V}|$, while maintaining the constraint that $E \subseteq \tilde{V} \times (V \cup \tilde{V})$.

Their method introduces fake vertices into the network and links them to each other and to real vertices in order to achieve the desired k-anonymity value. The authors introduced an $\mathcal{O}(kn)$ k-degree anonymization algorithm for unlabelled graphs based on dynamic programming and prove that, on any arbitrary graph, the minimisation of $|\tilde{V}|$ is optimal within an additive factor of k. For a special class of graphs that is likely to include social networks, the algorithm is optimal within 1 for reasonable values of k.

At a high level, the algorithm proceeds in three stages. First, Chester et al. designed a recursion to group the vertices of V by target degree (the degree they will have in $\tilde{G}$). The recursion establishes a grouping such that the *max deficiency*, a parameter in determining with how many vertices V must be augmented, is minimised. A dynamic programming with cost $\mathcal{O}(nk)$ is used to evaluate the recursion. The second stage is to determine precisely how many vertices with which we wish to augment V in order to guarantee that they can k-anonymize all of $\tilde{G}$. This number is a function of k and *max deficiency*. Finally, the algorithm introduces a particular means of adding new edges, each of which has at least one endpoint in $\tilde{G}$, with the objective of satisfying all the target degrees established during the recursion stage and k-anonymizing the new vertices added during the second stage. A critical property of this approach is that the

edges are added in such a manner as to guarantee tractability of the problem of k-anonymizing the new vertices, a problem which may be hard in the general case.

3.5 An evolutionary algorithm approach

Evolutionary Algorithm for Graph Anonymization[5] (in short, EAGA) [8] is a method focused on constructing a k-degree anonymous graph using evolutionary algorithms. A high-level description of this proposal allows us to structure it in two steps, in a similar way to the previous approaches.

Obtaining the k-degree anonymous sequence. In the first step, from the original degree sequence $d = \{d_1, \cdots, d_n\}$ of $G = (V, E)$, it constructs a new sequence $\tilde{d}$ which is k-degree anonymous and tries to minimize the distance Δ from the original sequence computed by Equation 1.

As we have commented, the anonymization of the degree sequence is computed by an evolutionary algorithm. The population is initialised from original degree sequence and many iterations are performed until a valid solution is found. The mutation process, which is responsible of the new candidates generation, applies a basic edge switch at each step (i.e, it adds one to an element of the sequence and subtracts one to another element of the sequence). This basic operation represents a change on a vertex of an edge, which is the most basic edge modification on a graph. For example, if an edge (v_i, v_k) is modified by replacing one vertex, one can obtain (v_j, v_k). This edge modification is represented on the degree sequence as a subtraction on vertex v_i (because it decreases its degree) and a addition on vertex v_j (because it increases its degree). It is important to note that this algorithm does not use crossover since this operation systematically breach the rule that preserves the number of edges of the graph, generating invalid candidates. The authors state the performance of the algorithm would be affected by the inclusion of this type of evolution and improvements would not occur in time or quality of the solution found. When candidate generation is done, the algorithm evaluates the candidates in order to find the best one. The score of each candidate is determined by the fitness function, which is a two-state function: if the k value of the candidate is lower than the desired one, the fitness function considers the dispersion in the degree histogram and the number of vertices which belong to groups between 0 and $k-1$ in the degree histogram, i.e, the number of vertices which does not fulfil de k-degree anonymity. This step is called "expansion" since the candidates tend to expand on the representation space trying to find a valid solution. On the contrary, if the k value of the candidate is equal or greater than the desired one, the fitness function only considers the distance from the original degree sequence. This step is called "retraction", since the candidates tend to move close to the original degree sequence. The candidate selection uses the steady-state model to choose the individuals which will survive to the next generation.

[5] Source code available at: http://deic.uab.cat/~jcasas/

Table 1: General properties of tested networks: number of vertices ($|V|$), number of edges ($|E|$), average degree ($\langle deg \rangle$) and default k-anonymity value (k).

Network	$\|V\|$	$\|E\|$	$\langle deg \rangle$	k
Polbooks	105	441	8.40	1
Polblogs	1,222	16,714	27.31	1

Modifying the original graph. In the second step, the algorithm constructs a graph $\tilde{G} = (\tilde{V}, \tilde{E})$ where $\tilde{V} = V$, $\tilde{E} \cap E \approx E$ and the degree sequence is equal to $\tilde{d}$. The difference between the anonymized and the original degree sequences $\tilde{d} - d$ points to vertices which have to increase or decrease their degree. Thus, some edges have to be added or removed from/to these vertices. The algorithm applies these modifications by edge switch, which consists on removing an edge $(v_i, v_k) \in E$, where v_i belongs to vertices which have to decrease their degree, and adding a new edge (v_j, v_k), where v_j belongs to vertices which have to increase their degree, as we show in Figure 3a.

4 Experimental Results

In this section we will compare the result of anonymizing processes using the four k-degree anonymous methods presented in Section 3. We apply all algorithms on the same data with the same parameters and compare the results in terms of information loss and data utility. We use several structural and spectral measures in order to quantify the information loss from distinct perspectives or network's characteristics. It is important to note that the privacy level is the same for all algorithms, as we compare results with the same k value. UMGA algorithm is applied using the neighbourhood centrality edge selection.

4.1 Tested networks

Table 1 shows a summary of the networks' main features, including number of vertices, number of edges, average degree and default k-anonymity value. US politics book data (polbooks) [24] is a network of books about US politics published around the 2004 presidential election and sold by the on-line bookseller Amazon. Edges between books represent frequent co-purchasing of books by the same buyers. Political blogosphere data (polblogs) [1] compiles the data on the links among US political blogs. Both of them are undirected and unlabelled networks.

4.2 Measures

In order to compare the algorithms, we use several well-known structural and spectral measures [36, 35, 4, 12]. The first structural measure is *harmonic mean of the shortest distance* (h). It is an evaluation of connectivity, similar to the

average distance or average path length. The inverse of the harmonic mean of the shortest distance is also known as the global efficiency, and it is computed by Equation 2, where $d(v_i, v_j)$ is the length of the shortest path from v_i to v_j, meaning the number of edges along the path.

$$\frac{1}{h} = \frac{1}{n(n-1)} \sum_{\substack{i,j=1 \\ i \neq j}}^{n} \frac{1}{d(v_i, v_j)} \tag{2}$$

Modularity (Q) indicates the goodness of the community structure. It is defined as the fraction of all edges that lie within communities minus the expected value of the same quantity in a network in which the vertices have the same degree, but edges are placed at random without regard for the communities.

Transitivity (T) is one type of clustering coefficient, which measures and characterizes the presence of local loops near a vertex. It measures the percentage of paths of length 2 which are also triangles.

Lastly, *sub-graph centrality* (SC) is used to quantify the centrality of vertex v_i based on the sub-graphs. Formally:

$$SC = \frac{1}{n} \sum_{i=1}^{n} SC_i = \frac{1}{n} \sum_{i=1}^{n} \sum_{k=0}^{\infty} \frac{P_i^k}{k!} \tag{3}$$

where P_i^k is the number of paths from v_i to v_i with length k.

Moreover, two spectral measures which are closely related to many network characteristics [36] are used. *The largest eigenvalue of the adjacency matrix A* (λ_1) where λ_i are the eigenvalues of A and $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$. The eigenvalues of A encode information about the cycles of a network as well as its diameter. *The second smallest eigenvalue of the Laplacian matrix L* (μ_2) where μ_i are the eigenvalues of L and $0 = \mu_1 \leq \mu_2 \leq \ldots \leq \mu_m \leq m$. The eigenvalues of L encode information about the tree structure of G. μ_2 is an important eigenvalue of the Laplacian matrix and can be used to show how good the communities separate, with smaller values corresponding to better community structures.

4.3 Empirical results

Results are disclosed in Table 2. Each row indicates the scored value for the corresponding measure and algorithm, and each column corresponds to an experiment with a different k-anonymity value. Each characteristic is reported from two to four times, corresponding to EAGA, UMGA, Liu and Terzi (indicated by L&T) and Chester et al. (indicated by Chester) algorithms. A bold row indicates the best algorithm for each measure and network. Values of Liu and Terzi algorithm are taken from Ying et al. [35] and values of Chester et al. algorithm are taken from [12]. Unfortunately, values for all measures and algorithms are not available. Perfect performance in a row would be indicated by achieving exactly the same score as in the original network (the $k = 1$ column). Although deviation is undesirable, it is inevitable due to the edge or vertex modification process.

Table 2: Results for EAGA, UMGA, Liu and Terzi (L&T) and Chester et al. (Chester) algorithms. For each dataset and algorithm, we vary k from 1 to 10 ($k = 1$ correspond to original dataset) and compare the results obtained on λ_1, μ_2, h, Q, T and SC. The last column correspond to the average error $\langle \mathcal{E} \rangle$. Bold rows indicate the algorithm that achieves the best results (i.e, lowest information loss) for each measure. Values of Liu and Terzi algorithm are taken from Ying et al. [35] and values of Chester et al. algorithm are taken from [12].

Polbooks		k=1	2	3	4	5	6	7	8	9	10	$\langle \mathcal{E} \rangle$
λ_1	EAGA	11.93	12.04	12.01	12.04	11.95	12.05	12.01	11.72	10.84	11.45	0.230
	UMGA		**12.09**	**11.97**	**11.85**	**11.85**	**11.95**	**12.09**	**12.08**	**12.08**	**11.86**	**0.090**
	L&T		12.00	12.05	12.11	12.22	12.30	12.31	12.64	12.72	12.85	0.383
μ_2	EAGA	0.324	0.372	0.477	0.496	0.516	0.515	0.600	0.595	0.578	0.321	0.156
	UMGA		**0.360**	**0.451**	**0.453**	**0.453**	**0.383**	**0.599**	**0.524**	**0.524**	**0.640**	**0.147**
	L&T		0.430	0.450	0.600	0.600	0.790	0.630	0.650	0.970	0.880	0.312
h	EAGA	2.450	2.378	2.324	2.346	2.297	2.314	2.294	2.282	2.308	2.421	0.109
	UMGA		**2.416**	**2.371**	**2.379**	**2.379**	**2.418**	**2.312**	**2.350**	**2.350**	**2.312**	**0.077**
	L&T		2.350	2.320	2.280	2.280	2.230	2.270	2.260	2.200	2.190	0.167
Q	EAGA	0.402	0.399	0.387	0.387	0.383	0.387	0.379	0.379	0.387	0.389	0.014
	UMGA		**0.400**	**0.393**	**0.396**	**0.396**	**0.401**	**0.386**	**0.386**	**0.386**	**0.385**	**0.009**
	L&T		0.390	0.390	0.380	0.380	0.360	0.370	0.370	0.340	0.350	0.027
T	EAGA	0.348	0.343	0.330	0.324	0.281	0.300	0.288	0.283	0.245	0.299	0.044
	UMGA		**0.350**	**0.342**	**0.339**	**0.339**	**0.347**	**0.326**	**0.322**	**0.322**	**0.324**	**0.013**
	L&T		0.330	0.330	0.320	0.330	0.300	0.310	0.320	0.290	0.300	0.023
$SC(\times 10^3)$	EAGA	2.524	2.624	2.333	2.293	1.751	2.001	1.967	1.415	0.653	1.534	0.634
	UMGA		**2.774**	**2.358**	**2.224**	**2.224**	**2.338**	**2.363**	**2.389**	**2.389**	**2.110**	**0.204**
	L&T		2.480	2.560	2.530	2.760	2.440	2.680	3.600	3.580	4.120	0.431
Polblogs		k=1	2	3	4	5	6	7	8	9	10	$\langle \mathcal{E} \rangle$
λ_1	EAGA	74.08	73.13	70.26	55.61	53.09	49.33	46.89	44.44	42.88	44.08	18.703
	UMGA		**73.93**	**73.81**	**73.92**	**73.95**	**73.74**	**73.80**	**73.75**	**73.63**	**73.61**	**0.256**
	L&T		74.89	74.50	75.16	75.10	76.32	75.82	76.67	77.42	78.42	1.758
μ_2	EAGA	0.168	0.168	0.168	0.692	0.674	0.748	0.754	0.690	0.858	0.757	0.517
	UMGA		0.168	0.168	0.168	0.168	0.168	0.168	0.168	0.168	0.168	0.000
	L&T		0.168	0.168	0.168	0.168	0.168	0.168	0.168	0.168	0.168	0.000
h	EAGA	2.506	2.677	2.623	2.596	2.592	2.588	2.595	2.565	2.572	2.575	0.071
	UMGA		**2.501**	**2.499**	**2.496**	**2.496**	**2.496**	**2.502**	**2.498**	**2.502**	**2.499**	**0.006**
	L&T		2.500	2.484	2.494	2.475	2.469	2.461	2.462	2.486	2.458	0.026
	Chester		2.506	2.486	2.476	2.476	2.456	2.456	2.446	2.436	2.426	0.039
Q	EAGA	0.405	0.404	0.404	0.402	0.395	0.399	0.401	0.392	0.400	0.397	0.005
	UMGA		**0.404**	**0.403**	**0.403**	**0.403**	**0.403**	**0.403**	**0.402**	**0.403**	**0.402**	**0.002**
	L&T		0.402	0.401	0.401	0.396	0.394	0.395	0.389	0.387	0.385	0.010
T	EAGA	0.226	0.224	0.219	0.148	0.130	0.110	0.104	0.086	0.078	0.082	0.085
	UMGA		**0.224**	**0.224**	**0.224**	**0.224**	**0.223**	**0.225**	**0.224**	**0.223**	**0.224**	**0.001**
	L&T		0.225	0.223	0.224	0.221	0.222	0.220	0.219	0.221	0.221	0.004
	Chester		0.219	0.215	0.207	0.205	0.200	0.226	0.190	0.185	0.183	0.020
$SC(\times 10^{29})$	EAGA	1.218	0.472	0.027	0.011	0.003	0.001	0.001	0.009	0.001	0.001	1.044
	UMGA		**1.052**	**0.932**	**1.040**	**1.068**	**0.871**	**0.921**	**0.875**	**0.776**	**0.765**	**0.266**
	L&T		2.730	1.870	3.610	3.400	1.450	6.940	6.250	4.460	4.040	2.386
	Chester		1.300	1.410	2.160	2.880	2.660	5.550	5.370	11.000	8.250	2.969

The first tested network, Polbooks, is a small collaboration network. We present values for EAGA, UMGA and Liu and Terzi algorithms. As shown in Table 2, UMGA algorithm introduces less noise on all measures. It outperforms on all measures, producing half of the average error in some measures, for example, λ_1 or SC. EAGA algorithm achieves the second best results on λ_1, μ_2, h and Q, while Liu and Terzi algorithm carry out on T and SC.

Polblog is the second tested network, which is considerably larger than the first one. Values for Chester et al. algorithm are presented for h, T and SC (other values are not available from Chester et al. [12]). Like in the previous test, UMGA algorithm gets the best values on all measures, except on μ_2 where Liu and Terzi algorithm achieves the same value. For instance, the average error is 0.006 for UMGA on h, while it rises to 0.026 for Liu and Terzi algorithm, 0.039 for Chester et al. approach, and 0.071 for EAGA. Similar results appear on λ_1, T and SC. Liu and Terzi algorithm obtains the second best results on λ_1, h and T, while EAGA does on Q and SC. Chester et al. approach by vertex addition gets values close to others algorithms, though the predictable level of information loss is slightly larger than the ones obtained by UMGA and Liu and Terzi algorithms. Despite the fact that EAGA gets good results on some metrics, the average error outbursts in many others. For example, results on λ_1 and μ_2 are larger than others, pointing out a considerable spectral noise introduced by the anonymization process.

We note two important factors which can be decisive for the quality of the anonymous data: The first one is the number of modifications in the edge and vertex set. Clearly, it is important to minimise these values since keeping them close to the original ones will preserve the structural and spectral metrics. The second factor we point out is related to edge relevance. Some edges play an important role inside the network, and preserving them we will lead the process to a better data utility and lower information loss. For instance, a bridge-like edge is critical for the structure of the network and the information flow. Thus, preserving it will conduct the anonymization process to a low information loss results. Notice that UMGA is the only algorithm which considers the edge relevance.

5 Conclusions

We have reviewed recent studies on anonymization techniques for privacy-preserving publishing of graph-formatted data. The research and development of privacy-preserving social network analysis is still in its early stage compared with much better studied privacy-preserving data analysis for tabular data. In this chapter we have focused on methods related to k-anonymity model, specifically to k-degree anonymity methods. These methods consider the vertices degree as adversary's knowledge, i.e, the adversary tries to re-identify a user in the anonymous data using the degree of some target vertices.

Four relevant methods of k-degree anonymity have been surveyed and compared. They are the algorithm by Liu and Terzi in [26], the approach using evolutionary algorithms and univariate micro-aggregation by Casas-Roma et al. in [8, 6], and the method based on vertex addition instead of only changing the edge set by Chester et al. in [12].

As we have stated before, the best results are achieved by the UMGA algorithm. We point out two important factors in order to reduce the information loss and preserve the data utility. Firstly, it is important to minimise the number of modifications in edge and vertex set, and secondly, considering the edge rel-

evance will reduce the noise in the anonymous data and preserve the structural and spectral properties.

Acknowledgements This work was partly funded by the Spanish Government through projects TIN2011-27076-C03-02 "CO-PRIVACY" and CONSOLIDER INGENIO 2010 CSD2007-0004 "ARES".

References

1. Adamic, L. A., and Glance, N. (2005). The political blogosphere and the 2004 U.S. election. In Int. Workshop on Link Discovery. NY, USA: ACM, pp. 36-43.
2. Backstrom, L., Dwork, C., and Kleinberg, J. (2007). Wherefore art thou r3579x? anonymized social networks, hidden patterns, and structural steganography. In Int. Conf. on World Wide Web. NY, USA: ACM, pp. 181–190.
3. Bhagat, S., Cormode, G., Krishnamurthy, B., and Srivastava, D. (2009). Class-based graph anonymization for social network data. Proc. of the VLDB Endowment, Vol. 2(1), pp. 766–777.
4. Boldi, P., Bonchi, F., Gionis, A., and Tassa, T. (2012). Injecting Uncertainty in Graphs for Identity Obfuscation. Proc. of the VLDB Endowment, Vol. 5(11), pp. 1376–1387.
5. Campan, A., and Truta, T. M. (2009). Data and Structural k-Anonymity in Social Networks. In Privacy, Security, and Trust in KDD. Springer-Verlag, pp. 33–54.
6. Casas-Roma, J., Herrera-Joancomartí, J., and Torra, V. (2013). An Algorithm For k-Degree Anonymity On Large Networks. In IEEE Int. Conf. on Advances on Social Networks Analysis and Mining. Niagara Falls, CA: IEEE, pp. 671–675.
7. Casas-Roma, J., Herrera-joancomartí, J., and Torra, V. (2013). Analyzing the Impact of Edge Modifications on Networks. In Int. Conf. on Modeling Decisions for Artificial Intelligence. Barcelona: Springer-Verlag, pp. 296–307.
8. Casas-Roma, J., Herrera-Joancomartí, J., and Torra, V. (2013). Evolutionary Algorithm for Graph Anonymization. ArXiv:1310.0229v2 [cs.DB], pp. 1–6.
9. Cheng, J., Fu, A. W., and Liu, J. (2010). K-isomorphism: privacy preserving network publication against structural attacks. In Int. Conf. on Management of Data. NY, USA: ACM, pp. 459–470.
10. Chester, S., Kapron, B. M., Ramesh, G., Srivastava, G., Thomo, A., and Venkatesh, S. (2011). k-Anonymization of Social Networks By Vertex Addition. In ADBIS 2011 Research Communications, pp. 107–116.
11. Chester, S., Gaertner, J., Stege, U., and Venkatesh, S. (2012). Anonymizing Subsets of Social Networks with Degree Constrained Subgraphs. In IEEE Int. Conf. on Advances on Social Networks Analysis and Mining. Washington, USA: IEEE, pp. 418-422.
12. Chester, S., Kapron, B. M., Ramesh, G., Srivastava, G., Thomo, A., and Venkatesh, S. (2013). Why Waldo befriended the dummy? k-Anonymization of social networks with pseudo-nodes. Social Network Analysis and Mining, Vol. 3(3), pp. 381–399.
13. Chester, S., Kapron, B. M., Srivastava, G., and Venkatesh, S. (2013). Complexity of social network anonymization. Social Network Analysis and Mining, Vol. 3(2), pp. 151–166.
14. Cormode, G., Srivastava, D., Yu, T., and Zhang, Q. (2010). Anonymizing bipartite graph data using safe groupings. The VLDB Journal, Vol. 19(1), pp. 115–139.

15. De Capitani di Vimercati, S., Foresti, S., Livraga, G., and Samarati, P. (2012). Data Privacy: Definitions and Techniques. Int. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, Vol. 20(6), pp. 793–818.
16. Dwork, C. (2006). Differential Privacy. Int. Conf. on Automata, Languages and Programming, Vol. 4052, pp. 1–12.
17. Hanhijärvi, S., Garriga, G. C., and Puolamäki, K. (2009). Randomization techniques for graphs. In SIAM Conf. on Data Mining. Nevada, USA: SIAM, pp. 780–791.
18. S. L. Hansen and S. Mukherjee. (2003). A Polynomial Algorithm for Optimal Univariate Microaggregation. IEEE Trans. on Knowledge and Data Engineering, Vol. 15(4), pp. 1043–1044.
19. Hay, M., Miklau, G., Jensen, D., Weis, P., and Srivastava, S. (2007). Anonymizing Social Networks, Technical Report 07-19, UMass Amherst, pp. 1–17.
20. Hay, M., Miklau, G., Jensen, D., Towsley, D., and Weis, P. (2008). Resisting structural re-identification in anonymized social networks. Proc. of the VLDB Endowment, Vol. 1(1), pp. 102–114.
21. Hay, M., Liu, K., Miklau, G., Pei, J., and Terzi, E. (2011). Privacy-aware data management in information networks. In Int. Conf. on Management of Data. New York, NY, USA: ACM, pp. 1201–1204.
22. Hay, M., Li, C., Miklau, G., and Jensen, D. (2009). Accurate Estimation of the Degree Distribution of Private Networks. In IEEE Int. Conf. on Data Mining. Miami, USA: IEEE, pp. 169–178.
23. Kapron, B. M., Srivastava, G., and Venkatesh, S. (2011). Social Network Anonymization via Edge Addition. In IEEE Int. Conf. on Advances on Social Networks Analysis and Mining. Kaohsiung: IEEE, pp. 155–162.
24. Krebs, V. (2006). http://www.orgnet.com.
25. Li, N., Li, T., and Venkatasubramanian, S. (2007). t-Closeness: Privacy Beyond k-Anonymity and ℓ-Diversity. In IEEE Int. Conf. on Data Engineering. IEEE, pp. 106–115.
26. Liu, K., and Terzi, E. (2008). Towards identity anonymization on graphs. In ACM SIGMOD Int. Conf. on Management of Data. New York, USA: ACM, pp. 93–106.
27. Machanavajjhala, A., Kifer, D., Gehrke, J., and Venkitasubramaniam, M. (2007). ℓ-diversity: Privacy beyond k-anonymity. ACM Transactions on Knowledge Discovery from Data, Vol. 1(1), pp. 3:1–3:12.
28. Narayanan, A., and Shmatikov, V. (2009). De-anonymizing Social Networks. In IEEE Symposium on Security and Privacy. Washington, DC, USA: IEEE, pp. 173–187.
29. Sihag, V. K. (2012). A clustering approach for structural k-anonymity in social networks using genetic algorithm. In CUBE Int. Information Technology Conference. Pune, India: ACM, pp. 701–706.
30. Samarati, P. (2001). Protecting Respondents Identities in Microdata Release. IEEE Transactions on Knowledge and Data Engineering, Vol. 13(6), pp. 1010–1027.
31. Stokes, K., and Torra, V. (2012). Reidentification and k-anonymity: a model for disclosure risk in graphs. Soft Computing, Vol. 16(10), pp. 1657–1670.
32. Sweeney, L. (2002). k-anonymity: a model for protecting privacy. Int. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, Vol. 10(5), pp. 557–570.
33. Tripathy, B. K., and Panda, G. K. (2010). A New Approach to Manage Security against Neighborhood Attacks in Social Networks. In IEEE Int. Conf. on Advances on Social Networks Analysis and Mining. Odense, Denmark: IEEE, pp. 264–269.

34. Wu, W., Xiao, Y., Wang, W., He, Z., and Wang, Z. (2010). K-symmetry model for identity anonymization in social networks. In Int. Conf. on Extending Database Technology. NY, USA: ACM, pp. 111–122.
35. Ying, X., Pan, K., Wu, X., and Guo, L. (2009). Comparisons of randomization and k-degree anonymization schemes for privacy preserving social network publishing. In Workshop on Social Network Mining and Analysis. NY, USA: ACM, pp. 10:1–10:10.
36. Ying, X., and Wu, X. (2008). Randomizing Social Networks: a Spectrum Preserving Approach. In SIAM Conf. on Data Mining. Atlanta, USA: SIAM, pp. 739–750.
37. Ying, X., and Wu, X. (2009). Graph Generation with Prescribed Feature Constraints. In SIAM Conf. on Data Mining. Sparks, USA: SIAM, pp. 966–977.
38. Zheleva, E., and Getoor, L. (2011). Privacy in Social Networks: A Survey. In C. C. Aggarwal (Ed.), Social Network Data Analytics. Springer, pp. 277–306.
39. Zhou, B., and Pei, J. (2008). Preserving Privacy in Social Networks Against Neighborhood Attacks. In IEEE Int. Conf. on Data Engineering. Washington, USA: IEEE, pp. 506–515.
40. Zhou, B., and Pei, J. (2011). The k-anonymity and ℓ-diversity approaches for privacy preservation in social networks against neighborhood attacks. Knowledge and Information Systems, Vol. 28(1), pp. 47–77.
41. Zou, L., Chen, L., and Özsu, M. T. (2009). k-Automorphism: A General Framework For Privacy Preserving Network Publication. Proc. of the VLDB Endowment, Vol. 2(1), pp. 946–957.