Social Network Analysis and Mining manuscript No.
(will be inserted by the editor)

Why Waldo Befriended the Dummy

k-Anonymization of Social Networks with Pseudo-Nodes

Sean Chester · Bruce M. Kapron · Ganesh Ramesh · Gautam Srivastava · Alex Thomo · S. Venkatesh

Received: date / Accepted: date

Abstract For a graph-based representation of a social network, the identity of participants can be uniquely determined if an adversary has background structural knowledge about the graph. We focus on degree-based attacks, wherein the adversary knows the degrees of particular target vertices and we aim to protect the anonymity of participants through k-anonymization, which ensures that every participant is equivalent to at least $k-1$ other participants with respect to degree. We introduce a natural and novel approach of introducing "dummy" participants into the network and linking them to each other and to real participants in order to achieve this anonymity.

The advantage of our approach lies in the nature of the results that we derive. We show that if participants have labels associated with them, the problem of anonymizing a subset of participants is *NP*-Complete. On the other hand, in the absence of labels, we give an $\mathcal{O}(nk)$ algorithm to optimally k-anonymize a subset of participants or to near-optimally k-anonymize all real and all dummy participants. For degree-based-attacks, such theoretical guarantees are novel.

Keywords privacy · k-anonymization · social networks · complexity · dynamic programming

S. Chester, B. M. Kapron, G. Srivastava, A. Thomo, S. Venkatesh
Department of Computer Science
University of Victoria
PO Box 3055 STN CSC
Victoria BC Canada V8W 3P6
E-mail: {schester,gsrivast}@uvic.ca, {bmkapron,thomo,venkat}@cs.uvic.ca

G. Ramesh
Yahoo! Inc.
4401 Great America Parkway
Santa Clara, CA 95054, USA
E-mail: rganesh@yahoo-inc.com

*A preliminary, short version [10] of this paper appeared at ADBIS 2011.

1 Introduction

The advent of the social web has brought about an explosive increase in the pervasiveness of large-scale social networks. Embedded within these networks is a wealth of information of multidisciplinary interest for the industrious analyst. However, unlike many other data, this information is about *people*; so, it is only ethical to respect their right to privacy. As a data owner, one must respect that some user groups expect a level of privacy and so that privacy must be provided to that subgroup if the data is to be shared with external organizations, if not the entire network. The study by Ferri et al. [19], for example, reveals that although some user groups are less concerned by data owners sharing data about them, up to 90% of members in other groups disagree with the principle.

If the social network is "sanitized" prior to release, the privacy of individuals within it can be protected while satisfying the needs of analysts. We assume the network is represented as a vertex-labelled graph $\mathcal{G} = (\mathrm{V}, \mathrm{E}, \Sigma, \ell)$, with vertices representing participants and edges representing the relationships between them.[1] As shown by Backstrom et al. [4], removing the identifiers from the graph is insufficient to guarantee privacy: an adversary can still reveal the identity of individuals by exploiting background structural knowledge that he has about the network. Figures 1(a) and 1(b) illustrate how degree, the structural property we study, can be sufficient to uniquely identify vertices: v is the only person with exactly 3 relationships, so stripping his identifier as in Figure 1(b) does not provide him any privacy against an adversary who knows his degree.

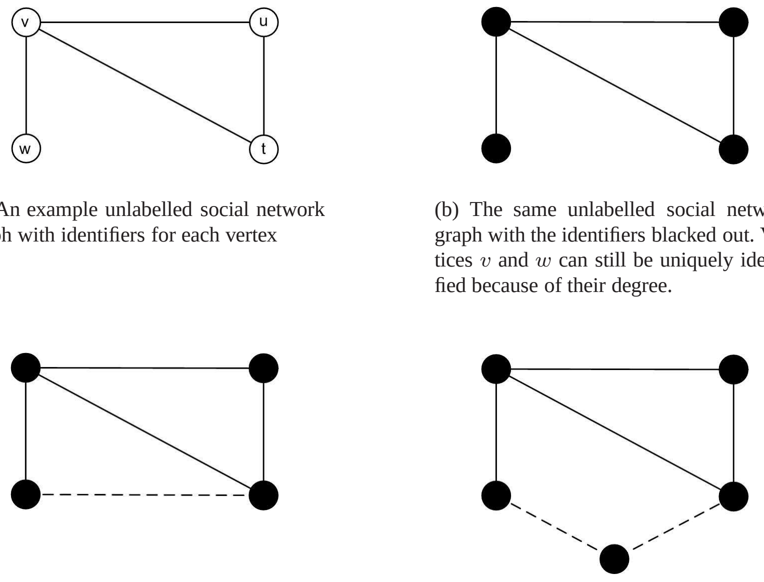
We pursue the well-known alternative of k-*anonymizing* the social network data prior to release. If an adversary has some knowledge of a structural property $\mathcal{P}$ of an arbitrary individual u who he seeks to identify, one can conceal the identity of u–and, indeed, everyone else–by establishing that every individual in the network is identical to at least $k-1$ others with respect to $\mathcal{P}$. In this way, the knowledge of the adversary renders at best a $1/k$ chance of uniquely identifying u. Stated informally, the k-$\mathcal{P}$-anonymizaton problem is thus: given a vertex-labelled graph $\mathcal{G}$ and a structural property $\mathcal{P}$, transform $\mathcal{G}$ into a k-$\mathcal{P}$-anonymous graph $\mathcal{G}'$ such that $\mathcal{G}'$ is as close as possible to $\mathcal{G}$. (Figures 1(c) and 1(d) are examples of k-$\mathcal{P}$-anonymous graphs for $\mathcal{P}$=degree.)

Certainly, there are many possible structural properties $\mathcal{P}$ of which the adversary may have some knowledge. Many of these have been studied in literature (e.g., [29, 43, 37, 39]) and appropriate k-anonymization techniques have been developed and shown to work well in experimental settings. Nonetheless, there are some pertitent, fundamental questions for any choice of $\mathcal{P}$:[2]

- Is it possible to produce an optimal k-$\mathcal{P}$-anonymous graph in polynomial time?

[1] We define a vertex-labelled graph as the four-tuple $(\mathrm{V}, \mathrm{E}, \Sigma, \ell)$, where V is a vertex set, $\mathrm{E} \subseteq \mathrm{V} \times \mathrm{V}$ is a set of undirected edges, Σ is a set of sensitive labels, and $\ell : \mathrm{V} \mapsto \Sigma$ is a labelling function that assigns a label to each vertex. We discuss in the paper two types of labels, sensitive and identifying. By Σ, we refer to the former, assuming the latter is stripped from the graph.

[2] We mention specific cases in which these questions have been answered in our discussion of related work in Section 6. Even in these cases, however, not all three questions have been fully addressed.



(a) An example unlabelled social network graph with identifiers for each vertex

(b) The same unlabelled social network graph with the identifiers blacked out. Vertices v and w can still be uniquely identified because of their degree.

(c) An optimal 2-degree-anonymization of the example graph by means of edge addition. No longer can vertices v and w be uniquely identified by degree.

(d) An optimal 2-degree-anonymization of the example graph by means of vertex addition. As before, vertices v and w cannot be uniquely identified by degree, but the method by which this property was achieved is different.

Fig. 1 An example of 2-degree-anonymity by means both of edge- (as in literature) and vertex- (as in this paper) addition.

- If not, can theoretical guarantees, such as bounds on approximation ratios, be derived for non-optimal algorithms?
- Can better theoretical guarantees be determined for special classes of graphs?

For any choice of adversarial knowledge, $\mathcal{P}$, these appear to be quite challenging problems. Consider the seemingly simple adversary of study in this work who knows only the degree of his target. Achieving *k-degree anonymity* to thwart him was studied by Lui and Terzi [29], who derived an experimentally effective algorithm. Meanwhile, the works of Chester et al. [11] and of Zhou and Pei [43] offer hardness results of k-$\mathcal{P}$-anonymization for other choices of $\mathcal{P}$. Still, none of these three most closely related works answer the questions above for k-degree-anonymization on arbitrary graphs, implying that the problems are still quite non-trivial for this adversary.

Thus, here, we introduce a new approach to k-$\mathcal{P}$-anonymization, changing both the way one goes about constructing $\mathcal{G}'$ and, correspondingly, the optimisation condition. In previous work, the problem formulation has been to transform $\mathcal{G} = (\mathrm{V}, \mathrm{E})$ into a k-anonymous graph $\mathcal{G}' = (\mathrm{V}, \mathrm{E} \cup \mathrm{E}')$; that is to say, only *edges* are added to $\mathcal{G}$ in order to construct $\mathcal{G}'$. (See Figure 1(c).) The optimisation condition, then, is to minimise $|\mathrm{E}'|$. We embark here on the very natural study of adding new *vertices* as well, transforming $\mathcal{G} = (\mathrm{V}, \mathrm{E})$ to $\mathcal{G}' = (\mathrm{V} \cup \mathrm{V}', \mathrm{E} \cup \mathrm{E}')$, and minimising $|\mathrm{V}'|$.[3] (See Figure 1(d).) For this optimisation problem, we require that the new edges must have a new vertex as an endpoint ($\mathrm{E}' \subseteq \mathrm{V}' \times (\mathrm{V} \cup \mathrm{V}')$); otherwise, the complete graph $\mathrm{K}_{|\mathrm{V}|}$ offers an optimal solution with $|\mathrm{V}'| = 0$, which is clearly not what we want. This alternative formulation presents notable utility. Ultimately, the intent of releasing data is to facilitate analysis, and the analysis is conducted at the aggregate level. Introducing few new vertices with similar characteristics to those already in the network could quite accurately preserve the aggregate characteristics of the network. Some network characteristics, such as the number of large cliques, are exactly preserved as a consequence of our constraint on E'.

Perhaps more importantly, however, is that by changing the problem formulation, we are able to address the aforementioned three problems for a degree-based attack. In particular, we prove hardness for k-anonymization against degree-based attacks on arbitrary, vertex-labelled graphs using this vertex-addition model (Theorem 2). By fixing the size of the label set to one (i.e., considering unlabelled graphs), we provide an efficient and optimal algorithm for subset anonymization (Corollary 2) and an efficient, near-optimal algorithm for complete anonymization (Theorem 1). Furthermore, for graphs with certain properties that are likely to arise in social networks, our optimality guarantee is improved to being within one vertex of optimal (Corollary 1).

Our Contributions

We introduce a vertex addition approach for k-$\mathcal{P}$-anonymization. For when $\mathcal{P}$ is degree, we offer the following results:

- We introduce for unlabelled graphs an $\mathcal{O}(k * |\mathrm{V}|)$ k-degree-anonymization algorithm based on dynamic programming and prove that, on any arbitrary graph, the minimisation of $|\mathrm{V}'|$ 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 (§ 3);
- We conduct an empirical evaluation of our algorithm on several well-known network datasets, demonstrating that it quite largely preserves the utility of the original graph with respect to standard structural parametres like clustering coefficient, average path length and connectivity, even as k approaches percentages of $|\mathrm{V}|$ that are quite high for the context of graph anonymization (§ 4);
- We demonstrate that for the more general case of labelled graphs ($|\Sigma| > 1$), k-degree-anonymization with a predetermined number of vertex additions (the decision version of the problem) is *NP*-Complete, by giving a reduction from a known *NP*-Hard table anonymization problem (§ 5).

[3] Precise formulations of the problem appear in Section **??** for unlabelled graphs and in Section 5 for labelled graphs.

2 Subset Anonymity and Related Concepts

We begin by introducing the concepts under study in this paper. At a high level, subset anonymity acknowledges that a certain subset of members in a social network may be highly identifiable as a group and, simultaneously, need an assurance of individual anonymity. For example, consider a social network constructed from a company email corpus where each node of the network corresponds to an email address and each link between two nodes implies an exchange of emails between them. The internal email addresses are likely to have much higher degree and be more sensitive than the external email addresses. The objective of subset anonymity is to ensure that members who belong to an identifiable subset of a social network, in this case the internal email addresses, cannot be distinguished from each other with certainty greater than $1/k$. This way, although they are identifiable as a group, they are not identifiable individually.

We model a social network as an unlabeled, undirected graph, $\mathcal{G} = (\mathrm{V}, \mathrm{E})$[4] which contains an *anonymizing subset* $\mathrm{X} \subseteq \mathrm{V}$ of vertices which need to be anonymized. The objective is to produce a similar graph $\mathcal{G}' = (\mathrm{V} \cup \mathrm{V}', \mathrm{E} \cup \mathrm{E}')$ in which X is anonymous. This differs from previous work, not only in the focus on subset anonymity, but also in the permission to introduce new, "fake" vertices (V').

An important consideration is what it means to be *anonymous* in a graph. Because graphs embed copious structural information, anonymity depends on what background structural information an attacker might have. As we detail in Section 6, various definitions have been introduced which assume an attacker has progressively more background structural information. Since no work, however, considers the implications of expanding the vertex set V, we embark on this *foundational* study in which we assume the simplest adversarial knowledge, that the attacker knows how many connections his target has, to contrast our conclusions with those derived from previous works which have not permitted an expansion of the vertex set (i.e., require $\mathrm{V}' = \emptyset$).

To state the problem formally, we first need to introduce a few definitions. We begin by explaining k-degree-anonymity, the focus of this paper:

Definition 1 The *degree* of a vertex v in a graph $\mathcal{G} = (\mathrm{V}, \mathrm{E})$ is the number of neighbours it has, $|\{u \in \mathrm{V} : (u, v) \in \mathrm{E}\}|$.

For example, the uppermost vertex in Figure 2(b) has a degree of 5, because it is connected to 5 other vertices (neighbours). The rightmost node in the same graph has degree 1 because it has only one incident edge. This is the information that we assume an attacker possesses.

Definition 2 A *degree sequence* of a set of vertices V is the sequence $(d_1, \ldots, d_n)$ composed by sorting in descending order the degrees of every node in V.

Again referring to Figure 2(b), the degree sequence is $(5, 3, 3, 2, 1, 1, 1)$, the degrees of each of the seven vertices sorted in descending order.

[4] For simplicity in this section, we regard a graph as a 2-tuple. We note that equivalently, for consistency, we could express an unlabelled graph as $\mathcal{G} = (\mathrm{V}, \mathrm{E}, \Sigma, \ell)$ where $\exists \sigma \in \Sigma : \forall v \in \mathrm{V}, \ell(v) = \sigma$. However, the simpler notation simplifies the exposition.

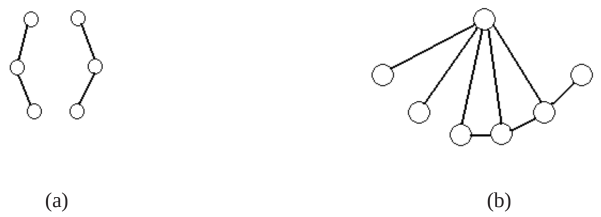


Fig. 2 The two small example graphs used to illustrate our anonymization procedure

Definition 3 A k*-partitioning* of a degree sequence is a partitioning of the degree sequence into disjoint partitions such that every degree appears in exactly one partition and every partition contains at least k elements.

Two possible 3-partitionings of the degree sequence in our running example are $((5,3,3),(2,1,1,1))$ and $((5,3,3,2),(1,1,1))$. There are other possible 3-partitionings of this degree sequence that have non-contiguous partitions, such as $((5,3,2,1),(3,1,1))$, but we will indicate later in Proposition 1 that these are never better choices for our algorithm.

Definition 4 A degree sequence is k*-anonymous* if partitioning it into the sets of distincts elements induces a k-partitioning.

That is to say, a degree sequence is k-anonymous if every degree appears at least k times in the sequence. Then, a subset of vertices can be defined as k-degree-anonymous depending on their degree sequence:

Definition 5 A subset of vertices $\mathrm{X} \subseteq \mathrm{V}$ is said to be k-degree-anonymous in a graph $\mathcal{G}' = (\mathrm{V}, \mathrm{E})$ iff the degree sequence of X is k-anonymous. Similarly, a *graph* is said to be k-degree-anonymous if its entire vertex set is k-degree-anonymous.

This provides sufficient material to define the problem of study here:

Problem Definition: *Given an input graph* $\mathcal{G} = (\mathrm{V}, \mathrm{E})$ *and an anonymizing subset* X*, construct a graph* $\mathcal{G}' = (\mathrm{V} \cup \mathrm{V}', \mathrm{E} \cup \mathrm{E}')$*,* $\mathrm{E}' \cap (\mathrm{V} \times \mathrm{V}) = \emptyset$*, such that* X *and* $\mathrm{X} \cup \mathrm{V}'$ *are both* k*-degree-anonymous in* $\mathcal{G}'$ *and* $|\mathrm{V}'|$ *is minimized.*

Note that X must be k-degree-anonymous in $\mathcal{G}'$ because it may be readily identifiable. This corresponds to a scenario when X is a small percentage of V and V' is indistinguishable from $\mathrm{V} \setminus \mathrm{X}$.[5] In other scenarios, say when users opt in to X as conceived by Yuan et al. [41], X could be a large percentage of V and not readily identifiable so we would need that $\mathrm{X} \cup \mathrm{V}'$ is k-degree-anonymous to hide the “fake” vertices among X rather than among $\mathrm{V} \setminus \mathrm{X}$. Thus the need for anonymity within both subsets.

[5] Considering the *Enron* email corpus on which we experiment in Section 4.1, $|\mathrm{V}| > 65000$, but only 151 vertices correspond to internal email addresses.

As a last note, we introduce a couple definitions that are useful for describing the algorithm in the next section.

Definition 6 The $\max \text{deficiency}$ of a k-partitioning of a degree sequence is the largest difference between highest and smallest degree within any partition.

For example, taking the two 3-partitionings in our running example, the $\max \text{deficiency}$ of $((5,3,3),(2,1,1,1))$ is $\max(5-3,2-1) = 2$ and of $((5,3,3,2),(1,1,1))$ is $\max(5-2,1-1) = 3$. Finally,

Definition 7 The total deficiency of a k-partitioning of a degree sequence is the sum over every d_i in the sequence of the difference between d_i and the largest degree in the same partition.

Using the same example we find that the total deficiency of $((5,3,3),(2,1,1,1))$ is:

$$((5-5)+(5-3)+(5-3)+(2-2)+(2-1)+(2-1)+(2-1)) = 7$$

and of $((5,3,3,2),(1,1,1))$ is:

$$((5-5)+(5-3)+(5-3)+(5-2)+(1-1)+(1-1)+(1-1)) = 7.$$

3 An Efficient Algorithm to Near-Optimally *k*-Degree-Anonymize Unlabelled Graphs

We present here immediately our most encouraging result, that for the special case of unlabelled graphs, k-degree-anonymization can be solved very near-optimally in linear time. We produce a k-degree-anonymous graph $\mathcal{G}' = (\mathrm{V} \cup \mathrm{V}', \mathrm{E} \cup \mathrm{E}')$ from an original, In $\mathcal{G}'$ we require that all the original vertices, V, are k-degree-anonymous-–and in Corollary 2, we will relax this constraint to an input subset $X \subseteq \mathrm{V}$. We 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 $\mathcal{G}$ (i.e., $\mathrm{V} \cup \mathrm{V}'$ is k-degree-anonymous in $\mathcal{G}'$). As mentioned in the introduction, we seek to minimise $|\mathrm{V}'|$, while maintaining the constraint that $\mathrm{E} \subseteq \mathrm{V}' \times (\mathrm{V} \cup \mathrm{V}')$. We will prove the following theorem at the end of this section:

Theorem 1. *Our algorithm produces a k-degree-anonymous graph $\mathcal{G}'$ containing the input graph $\mathcal{G}$ as an induced subgraph, using $\mathcal{O}(nk)$ time and $\mathcal{O}(n)$ space. The number of new vertices added is optimal up to an additive factor of k.*

At a high level, the algorithm proceeds in three stages. At first, we design a precise recursion to group the vertices of V by target degree (the degree they will have in $\mathcal{G}'$). The recursion establishes a grouping such that the $\max \text{deficiency}$, a parametre in determining with how many nodes V must be augmented, is minimised. We evaluate the recursion using dynamic programming with $\mathcal{O}(nk)$ execution cost.

The second stage is to determine precisely how many vertices with which we wish to augment V in order to guarantee that we can k-anonymise all of V'. This number is a function of k and $\mathrm{max\ deficiency}$, the parametre arising out of and minimised by the recursion evaluated in Stage 1.

Finally, we introduce a particular means of adding new edges, each of which has at least one endpoint in V', with the objective of satisfying all the target degrees established during the recursion of Stage 1 and k-anonymizing the new vertices added during Stage 2. A critical property of our specific approach is that the edges are added in such a manner as to guarantee tractibility of the problem of k-anonymizing the new vertices, a problem which may be hard in the general case.

As we describe the three stages of the algorithm in the following subsections, we illustrate their execution by 3-anonymizing the graphs in Figure 2.

3.1 Stage 1: Determining Each Vertex's Target Degree

Our algorithm first proceeds by identifying which vertices should have the same degree and what degree that should be. Similar to Liu and Terzi [29], we construct a recursion on the degree sequence of $\mathcal{G}$ to compute these groups and degrees. Contrary to their work, however, we minimise $\mathrm{max\ deficiency}$ rather than $\mathrm{total\ deficiency}$, because, as we prove later in Lemma 2, $\mathrm{max\ deficiency}$ tightly lower bounds the number of new vertices that must be added in order to k-anonymize the degree sequence of $\mathcal{G}$. Thus, we define a k-partitioning that minimizes $\mathrm{max\ deficiency}$ to be optimal.

To produce an optimal k-partitioning of the degree sequence of $\mathcal{G}$, we offer an incremental algorithm which operates from left (position 1) to right (position n) on the degree sequence, maintaining the optimal (i.e., with minimum $\mathrm{max\ deficiency}$) k-partitioning of those values in the degree sequence seen so far. The ability to do this with $\mathcal{O}(nk)$ cost is a consequence of the following propositions:

Proposition 1 *The* $\mathrm{max\ deficiency}$ *of a partition containing a highest degree of* d_i *and a smallest degree of* d_j *will be less or equal to the* $\mathrm{max\ deficiency}$ *of any partition containing* d_i *and any* d_{j+c} *or containing* d_j *and any* d_{i-c}, $\forall c \in \mathbb{N}$.

Proposition 2 *For any partition* $(d_i, \ldots, d_l)$*, its* $\mathrm{max\ deficiency}$ *is greater or equal to that of the partitions* $(d_i, \ldots, d_j)$, $(d_{j+1}, \ldots, d_l)$*, for* $i < j < l$*. That is to say, it never produces a higher* $\mathrm{max\ deficiency}$ *when one splits a partition.*

Both propositions follow from the facts that the degree sequences are sorted and $\mathrm{max\ deficiency}$ (i.e., difference) is transitive. Importantly, they allow us to construct a recursion, and an incremental, dynamic programming algorithm to evaluate that recursion, because they imply that there are only k ways to produce an optimal k-partitioning of the first x elements if the best possible k-partitionings are known for the first i elements, $\forall i < x$. From Proposition 1, it is clear that the x'th element should be added to the right of the first $(x-1)$ elements. From Proposition 2, it is clear that if there is an optimal split point for the rightmost partition that is farther than $2k-1$ positions left of x, then it can be split into sub-partitions such that the rightmost partition begins at some other position at least as far right as $x - 2k + 1$. The rightmost position must also have at least k elements; so, the rightmost partition of the optimal k-partitioning on the first x elements must begin between positions $x - 2k + 1$ and $x - k$, inclusive.

Our algorithm evaluates the recursion "bottom-up", constructing an optimal k-partitioning of the degree sequence by incrementally adding the next x'th rightmost degree and determining the best possible k-partitioning. If there are fewer than $2k$ degrees in the sequence, there is not any choice but to group them all together, because at least $2k$ elements are required to make two partitions of size $\geq k$. When, on the other hand, there are at least $2k$ degrees, then we evaluate the cost of splitting off a rightmost partition at any of the k positions between $x - 2k + 1$ and $x - k$, inclusive, and choose the rightmost of the cheapest among them. We invoke the recursion by recognising that the $\max \text{deficiency}$ incurred by creating a rightmost partition that starts at some position i is exactly the larger of the best possible k-partitioning up to $i - 1$ and the degree differences between the i'th and x'th degrees in the sequence.

The following recursion evaluates the cost of splitting and thus constructs an optimal k-partitioning. In the statement of the recursion below, the Δ function computes the $\max \text{deficiency}$ of a particular partition; the Start function keeps track of where x's partition starts, should x be the rightmost degree in it; and the Cost function computes the overall cost ($\max \text{deficiency}$) of the best possible partitioning up to the $x'th$ element. The Start function allows us to retrace the best possible k-partitioning up to any x'th position: the rightmost partition is given by $[\text{Start}(x), x]$; the next rightmost position is given by $[\text{Start}(\text{Start}(x) - 1), \text{Start}(x) - 1]$; &c. Finally, note that we assume argmin returns the maximal point at which a function is minimised.

Let

$$\begin{aligned} \text{Cost\_Split} &= \quad \min\nolimits_{i \in [\max(k, x-2k+1), x-k]}(\max(\text{Cost}(1, i-1), \Delta(i, x))), \\ \text{Pos\_Split} &= \text{argmin}_{i \in [\max(k, x-2k+1), x-k]}(\max(\text{Cost}(1, i-1), \Delta(i, x))). \end{aligned}$$

Then,

$$\begin{aligned} \Delta(x, y) &= d_x - d_y, \\ \text{Cost}(1, x) &= \Delta(1, x), \text{ if } x < 2k, \\ \text{Cost}(1, x) &= \text{Cost\_Split}, \text{ if } x \geq 2k, \\ \text{Start}(x) &= 1, \text{ if } x < 2k, \\ \text{Start}(x) &= \text{Pos\_Split}, \text{ if } x \geq 2k. \end{aligned}$$

For the graph of Figure 2(b), the degree sequence is $(5, 3, 3, 2, 1, 1, 1)$. The optimal k-partitioning of this degree sequence is $((5, 3, 3), (2, 1, 1, 1))$, which we arrive at by evaluating the recursion, as tabulated in Table 1.

Two parametres important for the next stages of our algorithm arise out of the k-partitioning of the degree sequence, namely $\max \text{deficiency}$ and total deficiency. As we show later in Lemma 2, the $\max \text{deficiency}$ is a lower bound on $|\text{V}'|$ in an

Pos	1	2	3	4	5	6	7
Deg Seq	5	3	3	2	1	1	1
$\mathrm{Cost}(1,x)$	0	2	2	3	4	2	2
$\mathrm{Start}(x)$	1	1	1	1	1	4	4

Table 1 The values of the recursion for the 3-degree-anonymization of the example graph from Figure 2(b)

optimal solution. We necessarily must bound the value of $\mathrm{total\ deficiency}$, because this allows us to upper bound the number of edges added by–and thus execution cost of–our entire algorithm.

Lemma 1 *The* $\mathrm{total\ deficiency}$ *of an optimal k-partitioning of a sorted degree sequence is upper-bounded by* $(n-1)(2k-1)$.

Proof First, note that from Proposition 2 that any optimal partitioning with a partition sized $2k$ or greater can be split into two partitions such that the $\max\mathrm{deficiency}$ is not increased. Furthermore, doing so will decrease the $\mathrm{total\ deficiency}$ unless the contribution of the partition is already zero. So, no optimal partition should contain $2k$ or more elements.

If $[\mathrm{front}(d_i), \mathrm{end}(d_i)]$ denotes the partition containing d_i, $[\mathrm{front}(p_j), \mathrm{end}(p_j)]$ denotes the partition p_j, and $|\mathcal{P}|$ denotes the number of partitions, then the total deficiency of an optimal k-partitioning is given by:

$$\begin{aligned}
&\sum_{d_i} \mathrm{front}(d_i) - d_i \\
\leq &\sum_{d_i} \mathrm{front}(d_i) - \mathrm{end}(d_i) \\
\leq &\ (2k-1) \sum_{p_j} \mathrm{front}(p_j) - \mathrm{end}(p_j) \\
\leq &\ (2k-1)\left(\mathrm{front}(p_1) - \mathrm{end}(p_{|\mathcal{P}|})\right) \\
\leq &\ (2k-1)(n-1),
\end{aligned}$$

where the second inequality follows because the deficiency contributed by a particular degree is certainly no more than the $\max\mathrm{deficiency}$ contributed by its entire partition, the third inequality follows because $\mathrm{front}(p_{j+1}) \leq \mathrm{end}(p_j)$, and the fourth inequality follows because the maximum degree in a simple graph is $n-1$.

3.2 Stage 2: Determining m, the Number of New Nodes

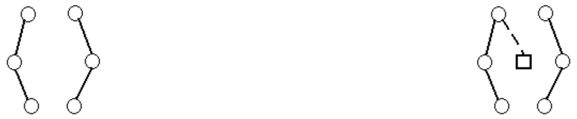
The next step in anonymizing a graph is to determine precisely how many vertices should be added. Ideally, we would add exactly $\max\mathrm{deficiency}$ vertices, because this is a lower bound on how many *must* be added, as shown in Lemma 2:

Lemma 2 *To k-degree-anonymize a set of vertices $X \subseteq \mathrm{V}$ by means of vertex addition, one must add at least as many new vertices as the* $\max\mathrm{deficiency}$ *of an optimal k-partitioning of the degree sequence of X.*

Proof First note that any graph $\mathcal{G}'$ in which X is k-degree-anonymous corresponds to some k-partitioning of the original degree sequence of X. In order to satisfy every target degree arising from the k-partitioning, some vertex will require as many new edges as the $\mathrm{max\ deficiency}$ of that k-partitioning. Also, because edges can only be added from each $v \in X$ to new vertices and because we investigate only simple graphs (i.e., those which do not contain multiple edges between the same source and destination nodes), clearly these new edges must connect to $\mathrm{max\ deficiency}$ new vertices in order to satisfy that particular vertex's requirement.

From among all k-partitionings of the degree sequence of X, corresponding to all graphs $\mathcal{G}'$ in which X is k-degree-anonymous, the optimal choice minimises $\mathrm{max\ deficiency}$, so also minimises how many new vertices are required in $\mathcal{G}'$.

However, for security, we anonymize the new vertices as well (ensure that $\mathrm{V} \cup \mathrm{V}'$ is k-degree-anonymous in $\mathcal{G}'$), and this may require adding more than $\mathrm{max\ deficiency}$ nodes. Our approach to adding edges we describe in more detail in the next subsection, but there are a couple important things to note here because they influence this stage. For readability, let $md = \mathrm{max\ deficiency}$ and let $td = \mathrm{total\ deficiency}$. Then, once every target degree is achieved through edge additions in Stage 3, $td (\mathrm{mod}\ md)$ of the new vertices will have some degree, call it d, and the other $md - td (\mathrm{mod}\ md)$ will have a degree of $d-1$. If these both appear in the target degrees of the k-partitioning, the entire graph $\mathcal{G}'$ is k-anonymous. Figure 3 illustrates an example of this, where the graph on the left is anonymized by adding one additional vertex which coincidentally then has a degree $d = 1$ that is already present on three other vertices. On the other hand, if d or $d-1$ does not appear in the target degrees, the new vertices need be explicitly anonymized as well. To accomplish this, we create a k-anonymous group of the new vertices by introducing intra-new-vertex edges to establish that they all have the same degree. This requires that: 1) there are k new vertices with which to form a k-anonymous group; and 2) there are an odd number of new vertices (we explain why in the next subsection).



(a) The example graph from Figure 2(a)

(b) A 3-degree-anonymization of (a) in which the new vertex ends up with the same degree as three vertices from the original graph, so does not need to be explicitly anonymized

Fig. 3 An example for which the additional vertex has the same degree as a 3-degree-anonymous group of original vertices

Both these conditions are satisfied when we add exactly

$$m = (1 + \max(md, k)) (\mathrm{mod}\ 2) + \max(md, k)$$

new vertices to the graph to create a new vertex set V' of size $n + m$.

3.3 Stage 3: The Edge Insertion

The final stage of our anonymization algorithm is to add new edges to the graph (restricting the addition to those with an endpoint in $\mathrm{V}' \setminus \mathrm{V}$, as per the problem definition), in order to meet the target degrees established in Stage 1. This must be done carefully because not any arbitrary approach is guaranteed to succeed: the task of anonymizing a subgraph of m new vertices added in Stage 2 is not generally trivial. Hence the motivation behind our *cycling* approach: it is designed such that it will always produces a scenario in which the anonymization of the new vertices is of linear cost.

First, we order the m additional vertices (arbitrarily). Let $def(v_i)$ be the discrepancy between the $i'th$ degree in the degree sequence and the largest degree within the same partition (the *deficiency* of the i'th vertex). We then connect the first $def(v_1)$ of the m additional vertices to the vertex corresponding to v_1, the next $def(v_2)$ additional vertices to the vertex corresponding to v_2, and so on until all m additional vertices have an edge. This process ends at some v_j.

We continue with another iteration, this time starting at v_j and with subsequent iterations until we have satisfied the deficiency of every node in the original graph. This is illustrated for the example graph of Figure 2(b) in the first five steps of Figure 4, reading left to right, then top to bottom. Seven edges must be added and they are done so cyclically from left to right.

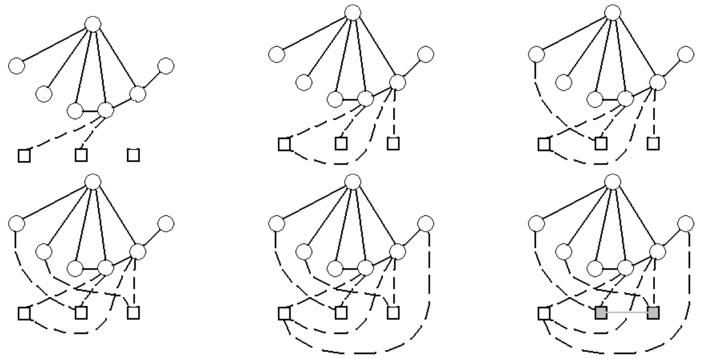


Fig. 4 3-Anonymizing the example graph from Figure 2(b) with three additional vertices

Because of the nature of this *cycling* procedure, always adding an edge to an additional vertex that has not yet been visited on that particular iteration, we can guarantee that the degrees of the additional vertices are all within one. In fact, if exactly d iterations are required in order to anonymize the original graph, then (as we hinted in Stage 2) $td(\bmod m)$ of the m additional vertices will have degree d and the remaining $m-td(\bmod m)$ will have degree $d-1$. Because we serviced each original vertex in turn, we can be certain to not accidentally introduce the same edge twice. So, we have successfully k-anonymized every vertex that was in the original graph. Quite importantly, we have done so in a manner that guarantees that the remaining edge-addition-based problem is efficient and successful because there is an odd number of vertices and the degrees are all within one.

The last detail is to ensure that all the new vertices are themselves k-anonymous. As a first recourse, if d and $d-1$ are both present as target degrees from the recursion in Stage 1, then the additional vertices will already belong to some k-anonymous group. Figure 3 demonstrates that 3-anonymizing the example graph of Figure 2(a) is such a case, where the additional vertex fits nicely into the 3-anonymous group of degree 1 vertices.

In the event that either d or $d-1$ is not present in the anonymized degree sequence, we explicitly anonymize the new vertices. For the $m - td(\bmod m)$ vertices with degree $d-1$, we randomly pair them and add an edge between each pair. If $m - td(\bmod m)$ is even and $m \geq k$, we know this is sufficient to guarantee all new vertices have the same degree (namely d) as at least $k-1$ other vertices.

If, instead, $m-td(\bmod m)$ is odd, then this pairing will leave out one last vertex, call it r. Because of our diligence in selecting m, we know that $m-1$ is even (and at least k), so we can add an edge from r to each of two other additional vertices so that all three have degree $d+1$. The remaining $m-3$ vertices with degree d can then all be paired off again (since $m-3$ is even) and all additional vertices will be anonymized with degree $d+1$. Some care must be taken to not accidentally re-add an edge between additional nodes, but this is really quite trivial because of how we proceeded with the preceding edge addition. Figure 5 illustrates this scenario. The degrees (4 and 3) of the two additional nodes are irreconcilable and $md-1$ is odd, so we instead our algorithm added the extra vertex as in Figure 4.

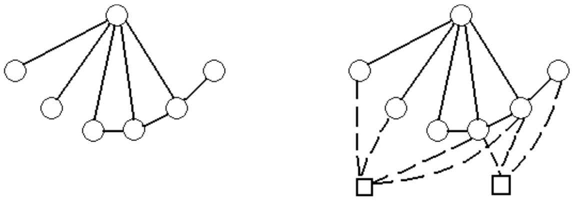


Fig. 5 An example in which our algorithm will not optimally 3-degree-anonymise Figure 2(b) with only two additional vertices because of the difficulty resolving the anonymity of the new vertices

3.4 Algorithmic Analysis

From the algorithm described in this section, we prove the following theorem:

Theorem 1 *Our algorithm produces a k-degree-anonymous graph $\mathcal{G}'$ containing the input graph $\mathcal{G}$ as an induced subgraph, using $\mathcal{O}(nk)$ time and $\mathcal{O}(n)$ space. The number of new vertices added is optimal up to an additive factor of k.*

Proof First, we note that the degree sequence partitioning requires $\mathcal{O}(n)$ space and $\mathcal{O}(nk)$ time. If this recursion is evaluated bottom-up (i.e., from $x = 1$ to $x = n$) and the results are memoised after each iteration, then the running time is linear in nk because the degree sequence has length n and for each element of the degree sequence there is an iteration that will cost at most a comparison to $2k$, lookups of Cost, and computations of Δ for each $i \in [k, 2k)$. The memory cost of the memoization is $3n$ units of memory: an array of size n each for storing the results of the Cost and Start calculations and an additional array of length n in which the original degree sequence is kept.

Second, the number of edges added is bounded by $td + m$, since td edges are added between original and additional vertices; the subsequent anonymization of additional vertices never changes an additional vertex's degree by more than $(d + 1) - (d-1) = 2$; and at least one additional vertex already has a degree of d. We introduce at most m additional vertices, at most either $md + 1$ or $k + 1$. k-Degree-anonymizing V implies, from Lemma 2, a lower bound on the optimal number of additional vertices is $md \geq 1$, noting also that any graph with a md of 0 is already n-anonymous. So, we add at most $md + 1 - md = 1$ more vertex than optimal or we add at most $k + 1 - 1 = k$ more vertices than optimal.

Since m is bounded by n (because neither md nor k can be larger than n) and the td is bounded by $(n - 1)(2k - 1)$, the addition of these edges requires $\mathcal{O}(nk)$ time. Since this requires constant memory, the overall space usage is $\mathcal{O}(n)$.

Note that we assume in this proof that the sorted degree sequence can be provided based on the representation of the graph. If this is not the case, an $\mathcal{O}(n \log n)$ preprocessing step to compute the sorted degree sequence subsumes our algorithm.

Corollary 1 *For input graphs in which $\exists i : (d_{i-1} - d_i) \geq k$ and $(d_i - d_{i+k-1}) \geq k$, our algorithm is optimal up to an additive factor of* 1.

Proof If $\exists i : (d_{i-1} - d_i) \geq k$ and $(d_i - d_{i+k-1}) \geq k$, then the $\max\text{deficiency}$ of the degree sequence is necessarily at least k because d_i must be grouped either with d_{i-1} or d_{i+k-1} by Proposition 1. Consequently, the scenario in which the algorithm adds some number of vertices other than $\max\text{deficiency}$ or $\max\text{deficiency} + 1$ does not arise.

This corollary is particularly interesting with respect to social network graphs, the degrees of which tend to follow a power law distribution [7].

3.5 On the Security of $\mathcal{G}'$

Perhaps an attacker's best chance at deconstructing our anonymization comes from cycling through all possible choices of the new equivalence classes (k-anonymous groups), removes them from the graph and then runs our algorithm to check if the output matches the original anonymized graph. We note here, that the last step where the adversary has to check if the output graph is the ”same” as the initial anonymized graph requires an algorithm for checking graph isomorphism. No efficient (polynomial time) algorithm is known for this problem. In addition, there are examples in which an equivalence class of additional vertices is merged with an equivalence class of original vertices because they have the same degree at the end of the anonymization procedure. For such graphs, this attack will not be able to extract the additional vertices only.

4 Experimental Evaluation of the Utility of $\mathcal{G}'$

Our algorithm, and indeed a vertex-addition approach in general, offers many advantages in terms of theoretical guarantees. In Section 3.4, for example, we proved asymptotic performance and near-optimality.

If we consider again the motivation for anonymizing the network in the first place, it is because ultimately we want to release it for analysis. In the case of an unlabelled graph, this is necessarily structural analysis. So, preserving the utility of the released data by not especially disrupting the structural characteristics of the graph is important. Thus more theoretical guarantees that our vertex-addition model offers: the number of cliques of size > 3 is exactly maintained between $\mathcal{G}$ and $\mathcal{G}'$. More broadly, for any monotone property of a graph, our approach can only possibly add false positives, never generating false negatives.

However, there are interesting non-monotone properties, too. In particular, for *clustering coefficient*, *average path length*, and the *hop plot* of a graph, the effect of any anonymization procedure is more difficult to predict and input-dependent, but the properties are focal points of other anonymization and social network analysis papers. In this section we present an experimental study in two parts of to what extent the application of our algorithm in the previous section disrupts the utility of the network with respect to these less predictable properties.

First, we investigate the scalability of the algorithm with respect to how well graph characteristics are preserved with increasingly demanding test scenarios (Section 4.1). Then, in Section 4.2 we investigate how well graph characteristics are preserved in the context of alternative approaches.

4.1 Scalability Tests

4.1.1 Metrics and Setup

Before describing the results of our experimental evaluation, we first detail our experimental setup. In particular, we describe the choice of datasets and metrics, and the implementation and machine details.

Datasets Four datasets from diverse domains form the subjects of our empirical study. We use one large dataset that represents an email communication network in a company (namely, Enron) [26]. Additionally, we select three other datasets that we expect to exhibit near worst-case behaviour. The *Net Science* graph [34] has a large discrepancy in degrees and thus incurs substantial $\mathrm{max\ deficiency}$ and $\mathrm{total\ deficiency}$ during the anonymization process. The *Prefuse* [23] and *Football* [20] graphs are small, so the effect of adding nodes and/or edges has a larger percentage effect on the properties of the graphs. The properties of these datasets are shown in Table 2. Note that Net Science, in particular, has been remarked to be a strong test set for network analysis by Leskovec et al. [27]. The Enron dataset was obtained from the Stanford SNAP repository,[6] and the Net Science and Football datasets were obtained from Mark Newman's repository.[7]

Graph	Nodes	Edges	APL	CC
Enron	36692	183831	3.39	0.09
Net Science	1589	2742	5.76	0.69
Prefuse	129	159	3.16	0.07
Football	115	613	2.51	0.61

Table 2 Structural Properties of Datasets for our Empirical Study

Metrics We measure the distortion introduced by the algorithm via some metrics which are commonly studied properties in the social network literature according to the survey by Chakrabarti et al. [7]. The three metrics we study are defined below.

1. Clustering Coefficient (CC) [5]: Informally, clustering coefficient measures the percentage of paths of length 2 which are also triangles. This metric in some sense measures *triadic closure* of graphs–social networks are known to have significant triadic closure (friends of a person are also likely to know each other). More formally, for all ordered triples $u, v, w \in V$,

$$CC = \frac{||\{u, v, w \in V : (u, v) \in E \wedge (u, w) \in E \wedge (v, w) \in E\}||}{||\{(u, v, w \in V : (u, v) \in E \wedge (u, w) \in E\}||}$$

2. Average Path Length (APL): This metric is a measure of the expected path length in the graph between any two randomly chosen connected vertices. This metric is highly relevant as it is directly related to the *six degrees of separation* that is known to exist between randomly chosen people in social networks, since the study of Milgram [33]. Define a predicate $C(u, v)$ to be *true* if u and v are connected in the graph and $false$ if they are not connected. Define $CP = \{(u, v) : C(u, v) = true\}$ to be the set of all the pairs of vertices that are connected. We define the average path length to be:

$$APL = \frac{\sum_{(u,v)\in CP} PathLength(u, v)}{\mid CP \mid}.$$

We assume that $PathLength(u, u) = 0$ for all $u \in V$.

[6] http://snap.stanford.edu/data/

[7] http://www-personal.umich.edu/~mejn/netdata/

3. Hop Plot [18]: The connectivity of a graph can be graphically modeled using the *hop plot*. The hop plot studies reachability for each path length k. For a given value of k, the hop plot displays, summed over all the vertices, the number of nodes reachable from that vertex using paths of length at most k. The maximum value for any value of k is n^2 where n is the number of vertices in the graph. The smallest value of k for which the maximum value of n^2 is reached is the *diameter* of the social network, the path length using which any two nodes in the graph can reach each other. Changing or distorting the connectivity of a graph drastically would change the shape of their hop plots. This is the main motivation behind studying these plots.

Experimental Setup A java implementation of the algorithm was used to measure the distortion based on the metrics for the five chosen datasets defined earlier in this section. The resulting graphs were manually verified to be k-anonymous. All experiments were performed on a quad-core Intel Xeon 5140 $2.33GHz$ processor with $4MB$ of L2 cache and $6GB$ of RAM.

Experimental studies in literature typically study small values of k (close to 3 and rarely even 100). We vary k as a fraction of n for our experiments, while still maintaining that $k \ll d$. For large and midsize datasets, we vary k from $k = 0.25$ up to 2% of the number of nodes in the graph. For the two small datasets (with over 100 nodes), this is too refined, so we vary k from $k = 1$ up to 5% of the number of nodes in the original graphs. For the large and midsize datasets, 2% of the number of nodes would still translate to k values of 80 to 720 which is a substantial number for the context.

4.1.2 Results and Discussion

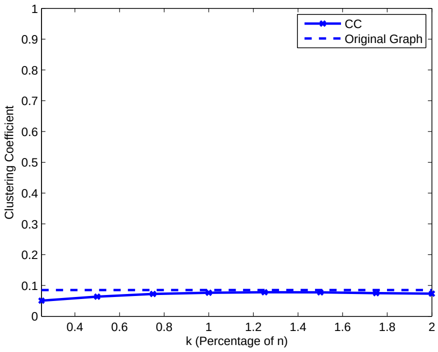


Fig. 6 Distortion of CC of Enron dataset as a function of k

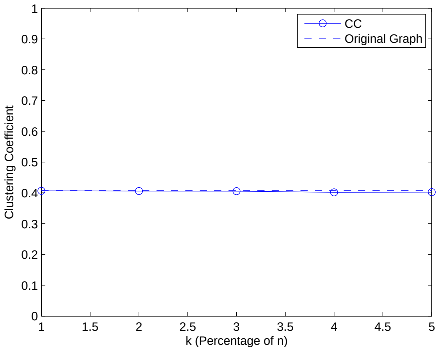


Fig. 7 Distortion of CC of Football dataset as a function of k

We first discuss the results of the experimentation on clustering coefficient. Notice first the plots for the Enron (Figure 6) and Football (Figure 7) datasets, which are examples of very good performance. The horizontal axis represents the value of k as a function of the number of nodes in the dataset. The vertical axis represents the clustering coefficient. Zero distortion would correspond to the solid line, which represents the characteristics of the anonymized data, exactly overlaying the dotted line, which represents the value in the original dataset. The performance of our algorithm on the Football dataset and for larger values of k on the Enron dataset is excellent because it very nearly achieves this perfect overlay.

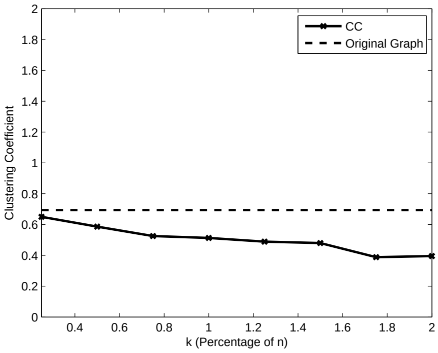


Fig. 8 Distortion of CC of Net Science dataset as a function of k

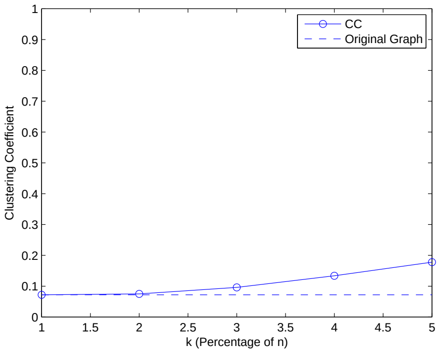


Fig. 9 Distortion of CC of Prefuse dataset as a function of k

Our performance with respect to clustering coefficient on the other two datasets, Net Science and Prefuse, is depicted in Figure 8 and Figure 9, respectively. That the opposite effect was observed on each dataset as k increased demonstrates the unpredictability of the effect of anonymization on clustering coefficient. In this case, the results match intuition, because of the properties of the dataset. In particular, Net Science has a high clustering coefficient originally; on the other hand, Prefuse features a couple nodes with especially high degree relative to the other nodes, so each newly added vertex is connected to a large percentage of original nodes. This is necessary in order to balance out the very high $td : md$ ratio at higher values of k.

Next, we discuss the results of the measurements on average path length. The plots that we have created are to be interpreted in the same manner as those just seen for clustering coefficient. That is to say, perfect performance occurs when the solid line, representing the anonymized graphs, overlays the dotted line, representing the value in the original graph. We again start by showing our performance on the Enron (Figure 10) and the Football (Figure 11) datasets, to demonstrate especially good results.

It is worth noting again the unpredictability of these metrics. The average path length on the Football dataset, unlike any of the others, rises in the anonymized graph. We note that the only especially large difference between the original graph and anonymized graph occurs on the Net Science dataset (Figure 12), which has a relatively high original average path length compared to the other datasets. Although

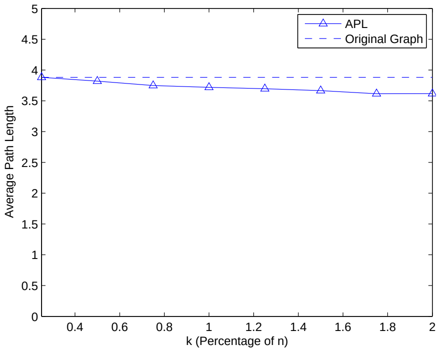


Fig. 10 Distortion of APL of Enron dataset as a function of k

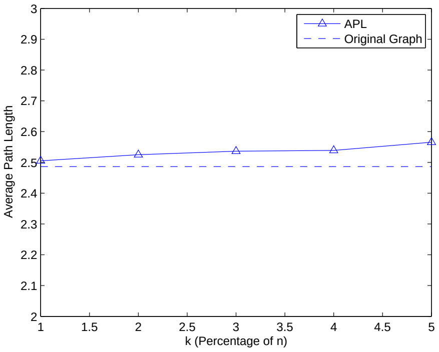


Fig. 11 Distortion of APL of Football dataset as a function of k

there is a noteable change on the Prefuse dataset (Figure 13) for $k = 2\%$, it decreases at a very slow rate for all subsequent values $k > 2\%$.

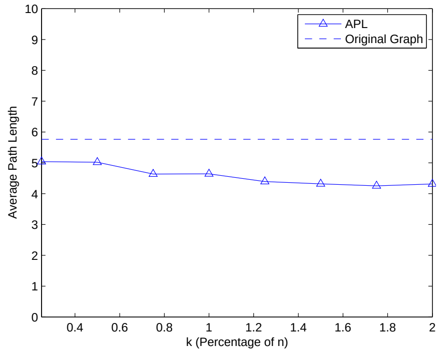


Fig. 12 Distortion of APL of Net Science dataset as a function of k

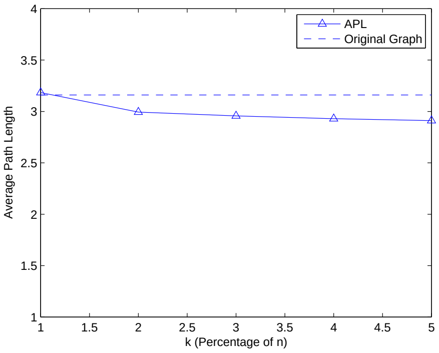


Fig. 13 Distortion of APL of Prefuse dataset as a function of k

The final set of experiments that we ran on the structural distortion of our algorithm for non-monotone properties was in measuring the hop plot of the graphs. The structure of these plots is different, but the interpretation is similar. For each value of k, a separate line is depicted. Additionally, one line depicts the hop plot for the original graph. Because the number of vertices changes, the lines cannot possibly overlay each other. Instead, good results are indicated by the similarity of the *shape* of the curves. The results on the Enron (Figure 14), Football (Figure 15), and Prefuse (Figure 17) datasets demonstrate especially good performance, because the shape of the curve is consistent despite its increasing max value.

Notice that the hop plot lines for the Football dataset nearly *do* overlay each other: this is because the number of additional nodes m required to anonymize the dataset

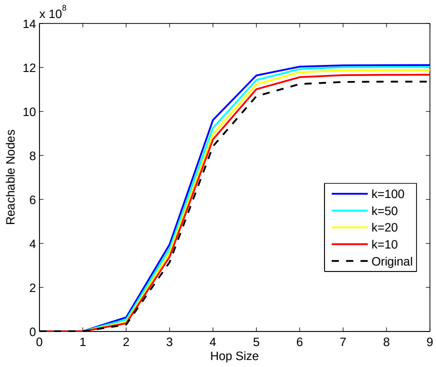


Fig. 14 Distortion to the hop plot of Enron dataset for several values of k

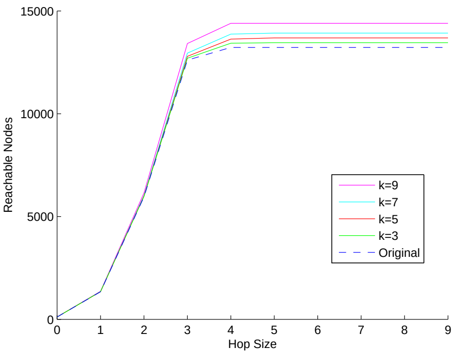


Fig. 15 Distortion to the hop plot of Football dataset for several values of k

is quite small. The Net Science dataset (Figure 16) has a hop plot that varies more substantially after anonymization. Again, however, most of this change occurs at a low value of k; for subsequent, higher values, not much more change is witnessed.

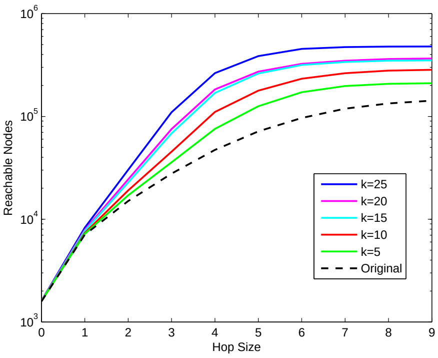


Fig. 16 Distortion to the hop plot of Net Science dataset for several values of k

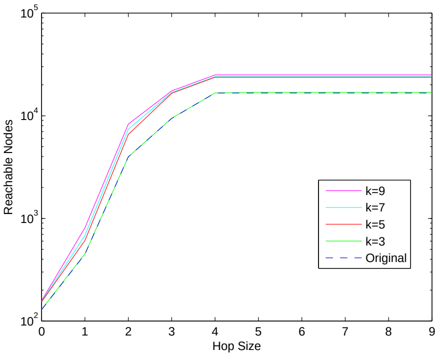


Fig. 17 Distortion to the hop plot of Prefuse dataset for several values of k

A last comment from our scalability evaluation is with respect to execution time. Indeed, we proved good asymptotic performance, but, nonetheless, absolute running time is always a curious measure. For the largest dataset (Enron), the running time for the actual anonymization took a little over one minute (70 seconds) for all the values of k in the plots, reflecting the proven efficiency of the algorithm. The running time reported is the average of 5 independent runs of the algorithm for each value of k. The evaluation time was dominated by the naive computation of the metrics on the original and distorted graphs. These took about 20 to 30 minutes each. The times on smaller graphs were much lower and had the same trend where the computation of the metrics dominated the running time.

4.2 Comparability Tests

Although there are . Compare to the state-of-the-art edge-based k-degree-anonymization algorithm [29]. Evaluate whether the vertex-based approach that we introduce is empirically competitive in terms of what is known about the edge-based approach.

4.2.1 Methodology

An experiment was run earlier by Ying et al. [40] to compare the k-degree-anonymization of Lui and Terzi [29] to an approach that randomly adds and removes edges to provide anonymity through obscurity. Our methodology for comparing to the algorithm of Lui and Terzi is to run a series of experiments identical those of Ying et al. and compare the performance of our algorithm to these published values. Thus, as before, we anonymize the entire vertex set, V.

Datasets We conduct the experiments on the same *polblogs* dataset [1] used by Ying et al., which we obtained from the Carnegie Mellon CASOS repository[8]. The dataset consists of 1222 vertices, each corresponding to a political blog b_i, and 16714 edges (b_i, b_j) indicating that a hyperlink existed from b_i to b_j and/or from b_j to b_i. In other words, the dataset is an undirected citation network.

Metrics Ying et al. evaluate four metrics that were presented in the comprehensive review of network characteristics by Costa et al. [13]. We evaluate three of these; the modularity measure, Q, on the other hand, does not apply in our case because we assume vertices are all of the same type (i.e., unlabeled), and it evaluates the extent to which the network is modularized by vertex type.

1. Transitivity Measure (C) [5]: The transitivity measure used by Ying et al. is exactly the *clustering coefficient* that we introduced in Section 4.1. We will continue to refer to this as the clustering coefficient of the graph.
2. Harmonic Mean of the shortest path (h) [25]: The *harmonic mean* is an evaluation of connectivity, similar to the *average path length* that we used in Section 4.1. Let d_{ij} be the distance of the shortest path from vertex i to vertex j, or ∞ if they not connected. Then,
$$\frac{1}{h} = \frac{1}{|\mathrm{V}|(|\mathrm{V}|-1)} \sum_{i \neq j} \frac{1}{d_{ij}}.$$
3. Subgraph Centrality (SC) [17]: The *subgraph centrality* is an evaluation of how many (not necessarily simple) cyles emanate, on average, from each vertex, where each cycle is weighted by the reciprocal of the factorial of its length. Let W_i^l be the number of walks[9] of length l that start and end at vertex i. The subgraph

[8] `http://www.casos.cs.cmu.edu/computational_tools/datasets/external/polblogs/index11.php`

[9] Recall that a walk is any sequence of adjacent edges, including those which revisit edges and/or vertices.

centrality is the following convergent infinite summation, which converged to the precision we report in these experiments at a length of $l \approx 120$:

$$SC = \frac{1}{|V|} \sum_{i=1}^{|V|} \sum_{l=2}^{\infty} \frac{W_i^l}{l!}.$$

Experimental Setup The experiments are conducted identically to those in our scalability tests (Section 4.1), except that we instead vary k through integers $k \in [2, 10]$, for precise compatibility with the study by Ying et al.We report all values to the same precision as offered by Ying et al.

4.2.2 Results

The results for the comparability study are disclosed in Table 3 below. Each row of the table corresponds to an experiment with a different anonymity threshold. Each column indicates the measured value for the corresponding graph characteristic. Each characteristic is reported twice, once corresponding to an edge-addition algorithm and indicated by the suffix *-E* and once corresponding to our node-addition algorithm and indicated by the suffix *-V*. Perfect performance in a row would be indicated by achieving exactly the same characteristics as in the original graph (the $k = 1$ row); deviation in either direction is undesirable.

					SC-V	**SC**-E
	CC-V	**CC**-E*	**h**-V	**h**-E*	($\times 10^{29}$)	($\times 10^{29}$)
k=1	**0.226**	**0.226**	**2.51**	**2.51**	**1.21**	**1.21**
k=2	0.219	0.225	2.51	2.50	1.30	2.73
k=3	0.215	0.223	2.49	2.48	1.41	1.87
k=4	0.207	0.224	2.48	2.49	2.16	3.61
k=5	0.205	0.221	2.48	2.48	2.88	3.40
k=6	0.200	0.222	2.46	2.47	2.66	1.45
k=7	0.226	0.220	2.46	2.46	5.55	6.94
k=8	0.190	0.219	2.45	2.46	5.37	6.25
k=9	0.185	0.221	2.44	2.49	11.0	4.46
k=10	0.183	0.221	2.43	2.46	8.25	4.04

*Values for edge-addition (E) taken from [40].

Table 3 Values of *clustering coefficient* (CC), *harmonic mean* (h), and *subgraph centrality* (SC), respectively, for the *polblogs* dataset after applying k-degree-anonymization by means of vertex addition (V), as in this paper or strictly edge addition (E), as in [29]. The row $k = 1$ represents the original characteristics of the graph (i.e., only a guarantee of 1-anonymity). Note that the scale is changed for the two columns reporting *subgraph centrality*.

4.2.3 Discussion

The results of the comparability experiments are very encouraging. Our algorithm does not rely on any randomization, yet still performs very near to the edge-addition algorithm on the first two attributes. For *clustering coefficient* our algorithm is within

0.018 of the edge-additions values, on average, with a standard deviation across points of 0.015. This is very close, considering the precision of the experiments is reported only to within 0.001. For *harmonic mean*, these numbers are even closer: $\mu = 0.01$ and $\sigma = 0.02$ with an experimental precision of 0.01.

These values are closer even than they seem because of the randomization that is involved in the edge-addition algorithm. Whereas our algorithm will always produce the same values for these characteristics, any values reported for the algorithm of Lui and Terzi are subject to randomness and could be either higher or lower on any subsequent run of the experiments. This suggests that either approach to k-anonymization can produce anonymous graphs that have similar structural characteristics.

Results for the final metric under consideration, *subgraph centrality*, are somewhat inconclusive. Our algorithm outperforms the edge-addition on all but one case prior to $k = 9$. This is reasonable to expect, because adding edges to existent nodes is more likely to create new, short loops than is adding edges to "fake" nodes and the addition of new nodes by our algorithm increases the denominator to help stunt the growth of the numerator. For the highest values of k, 9 and 10, the edge-addition has an unintuitive 40% leap in performance in contrast to our predictable degradation. Consequently, it is difficult to extrapolate what might occur with this metric in new experiments.

5 A Hardness Result for Subset Anonymization on Labelled Graphs

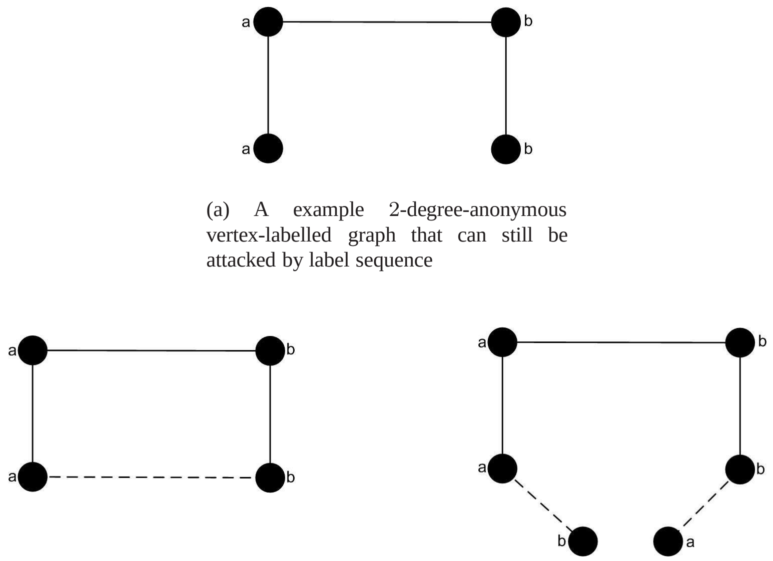
Until this point, we have been considering the special case of unlabelled graphs. For example, we stripped the political lean labels from the *polblogs* dataset during our experiments. In this section, we increase $|\Sigma|$ and study the tractability of k-anonymization against degree-based attacks. With $|\Sigma| > 1$, the nature of a degree-based attack changes, because the adversary may know the number of outbound edges from a target v to vertices of each label. So, a stronger anonymity notion is required. (See Figure 18.) Also, we recognize that often it is not the case that *every* vertex need be anonymous (for example, see the work of Yuan et al. [41]), so consider *subset anonymization*. We begin by formally defining the problem for the labelled setting. Note that for $|\Sigma| = 1$, this reduces to the problem introduced in §3.

The analogue in the labelled setting of a degree sequence is a label sequence:

Definition 8 (Label sequence) For $v \in \mathrm{V}$, we say that $S_v = (l_1, l_2, \ldots, l_m)$ is a label sequence for v if it corresponds to some ordering of the labels of v and the vertices that are adjacent to v. We will consider label sequences of vertices to be equivalent up to reordering.

The four label sequences in the example of Figure 18(a), anticlockwise from the bottom left, are $((v, v), (v, v, u), (v, u, u), (u, u))$. Then, k-anonymity for labelled graphs relates to the uniqueness of label sequences:

Definition 9 Given a vertex-labelled graph $\mathcal{G} = (\mathrm{V}, \mathrm{E}, \Sigma, \ell)$, a subset $X \subseteq \mathrm{V}$ of vertices is k-sequence-anonymous in $\mathcal{G}$ if for every vertex $v \in X$, there are at least $k - 1$ other vertices in X whose label sequence is the same as the label sequence of v.



(a) A example 2-degree-anonymous vertex-labelled graph that can still be attacked by label sequence

(b) The example vertex-labelled graph made to be 2-sequence-anonymous by means of *edge* addition

(c) The example vertex-labelled graph made to be 2-sequence-anonymous by means of *vertex* addition

Fig. 18 An example of 2-sequence-anonymity. Notice that 2-degree-anonymity (Figure 18(a)) is insufficient to guarantee anonymity in a vertex-labelled setting.

We denote by $u \equiv v$ that vertices u and v have the same label sequence. Clearly, $\equiv$ is an equivalence relation and hence induces a partition $X/\equiv$ of X into equivalence classes. If X is k-anonymous in $\mathcal{G}$, every equivalence class is of size at least k. Then, the problem of k-sequence anonymization is defined as:

k-LABELLED SUBGRAPH ANONYMIZATION
Input: A vertex-labelled graph $\mathcal{G} = (\mathrm{V}, \mathrm{E}, \Sigma, \ell)$, a set $X \subseteq \mathrm{V}$ of vertices, and an integer t.
Question: Is there a vertex-labelled graph $\mathcal{G}' = (\mathrm{V} \cup \mathrm{V}', \mathrm{E} \cup \mathrm{E}', \Sigma \cup \Sigma', \ell \cup \ell')$ such that $|\mathrm{V}'| \leq t$, $\mathrm{E}' \subseteq (\mathrm{V} \times \mathrm{V}') \cup (\mathrm{V}' \times \mathrm{V}')$, $\Sigma'_{|\mathrm{V}} = \Sigma$, $\ell'_{|\mathrm{V}} = \ell$ and X is k-anonymous in $\mathcal{G}'$?

That is, can we k-sequence-anonymize X by adding at most t new labelled vertices? New edges must have at least one endpoint in V'.

Theorem 2 *k-Labelled Subgraph Anonymization is NP-Complete for $k \geq 3$.*

Proof To show that this problem is *NP*-Hard, we build a reduction from the following anonymization problem on tables that was shown to be *NP*-Hard by Meyerson and Williams [32]:

k-ATTRIBUTE-ANONYMITY

Input: A table T with n rows and l columns (also called attributes) with entries over $\{0,1\}$ and integers t and $k \geq 3$.
Question: Can the rows of T be k-anonymized by suppressing at most t attributes of T? An attribute is said to be suppressed if all its entries (0 and 1) are replaced by *.

Reduction: Our reduction is described as follows:

Given a Table T, let $\mathrm{T}_{(m,j)} \in \{0,1\}$ denote the value of attribute j in row m. Then, the vertex-labelled graph $\mathcal{G}_{\mathrm{T}}$ corresponding to T is constructed as follows:

– $\mathrm{V}_{\mathrm{T}} = \{r_1, r_2, \ldots, r_n\} \cup \{c_j^i | 1 \leq j \leq l, i \in \{0,1\}\}$.
– Let $\mathrm{E}_{\mathrm{T}} = \{(r_m, c_j^i) | \mathrm{T}_{(m,j)} = i\}$ where $1 \leq m \leq n$, $1 \leq j \leq l$ and $i \in \{0,1\}$.
– $\Sigma = \{R\} \cup \{A_j^i | 1 \leq j \leq l, i \in \{0,1\}\}$.
– $\ell(r_i) = R$ and $\ell(c_j^i) = A_j^i$.
– Finally, remove all isolated vertices from $\mathcal{G}_{\mathrm{T}}$.

Entry	A_1	A_2	A_3
1	0	0	1
2	0	1	1
3	0	0	1
4	1	0	1
5	1	0	0
6	1	0	1

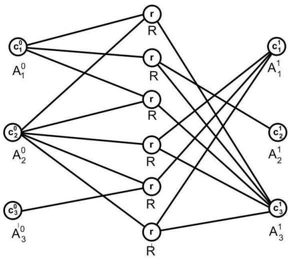


Fig. 19 An example table T and its tranformation into a graph $\mathcal{G}_{\mathrm{T}}$

In other words, we encode a binary table as a graph in which a row vertex r_m with label R is connected to a column vertex c_j^0 (c_j^1) with label A_j^0 (A_j^1) if the (m,j)th entry is 0 (1).

Let $X = \{r_1, \ldots, r_n\}$ denote the set of row vertices of $\mathcal{G}_{\mathrm{T}}$. We will show that T can be k-anonymized by suppressing at most t attributes if and only if we can k-anonymize X by adding at most $2t$ new labelled vertices.

Let $\mathcal{G}'_{\mathrm{T}}$ be any graph obtained from $\mathcal{G}_{\mathrm{T}}$ such that X is k-anonymous in $\mathcal{G}'_{\mathrm{T}}$. We will now show that we may assume without loss of generality, $\mathcal{G}'_{\mathrm{T}}$ satisfies a set of properties. If it does not, we convert it into one that satisfies the properties without increasing the number of new labelled vertices added in $\mathcal{G}'_{\mathrm{T}}$.

Our first lemma shows that the anonymization procedure does not introduce vertices with new labels not in Σ_{T}. Furthermore, none of the new vertices added to $\mathcal{G}_{\mathrm{T}}$ will be labelled by R.

Lemma 3 *$\Sigma'_{\mathrm{T}} = \Sigma_{\mathrm{T}}$ and $\ell'_{\mathrm{T}}(v) \neq R$ for any $v \in \mathrm{V}'_{\mathrm{T}}$.*

Proof Suppose there is at least one vertex with a new label l, $l \notin \Sigma_{\mathrm{T}}$ in $\mathcal{G}'_{\mathrm{T}}$. Let Y be any equivalence class of $X/\equiv$. Every vertex $y \in Y$ has the same number of labels l in its label sequence in $\mathcal{G}'_{\mathrm{T}}$. Therefore, the label sequences of all vertices in Y will remain the same after all the new vertices with label l are removed from

$\mathcal{G}'_{\mathrm{T}}$. Since this is true of any equivalence class Y, we can assume that $\Sigma'_{\mathrm{T}} = \Sigma_{\mathrm{T}}$. To show that $\ell'_{\mathrm{T}}(v) \neq R$ for any $v \in \mathrm{V}'_{\mathrm{T}}$, note that the label R does not appear in the label sequence of any vertex of Y in $\mathcal{G}_{\mathrm{T}}$. In addition, R must appear the same number of times in the label sequence of any vertex of Y in $\mathcal{G}'_{\mathrm{T}}$. Therefore, by the same reasoning as above, all new nodes with the label R in $\mathcal{G}'_{\mathrm{T}}$ can be removed and Y will remain k-anonymous.

Lemma 4 *For every i, $i \in \{0, 1\}$, and every j, $j \in \{1, 2, \ldots, l\}$, the label A^i_j appears at most once in the label sequence of a vertex from X in the graph $\mathcal{G}'_{\mathrm{T}}$.*

Proof Let Y be any equivalence class of X. All vertices in Y have the same label sequence in $\mathcal{G}'_{\mathrm{T}}$. If a label A^i_j appears in the label sequence of a vertex y of Y in $\mathcal{G}_{\mathrm{T}}$, we can assume that no edge between y and a new vertex labelled A^i_j is added in $\mathcal{G}'_{\mathrm{T}}$. If not, this label will appear more than once in the label sequence of y and hence in the label sequence of every other vertex of Y. Also, the number of occurrences of this label in every label sequence will the same. Since at most one of these will be due to an edge of $\mathcal{G}_{\mathrm{T}}$, for each vertex v in Y, we can remove all but one edge in $\mathrm{E}_{\mathrm{T}} \cup \mathrm{E}'_{\mathrm{T}}$ between v and a node labelled A^i_j in $\mathcal{G}'_{\mathrm{T}}$ and still preserve anonymity of vertices in Y.

Lemma 5 *For every i, $i \in \{0, 1\}$, and every j, $j \in \{1, 2, \ldots, l\}$, at most one new vertex with the label A^i_j is added in the graph $\mathcal{G}'_{\mathrm{T}}$.*

Proof Suppose not. We merge all the new nodes with the label A^i_j into a single node. This only reduces the number of new vertices added with label A^i_j. In Lemma 4, we have shown that every label A^i_j appears at most once in the label sequence of every vertex in X. Therefore, the merge operation will not create multiple edges and all the label sequences of the vertices of X will remain the same as before after this modification.

Lemma 6 *All new vertices appear in pairs. That is, if $A^0_j \in \ell(\mathrm{V}'_{\mathrm{T}})$, then $A^1_j \in \ell(\mathrm{V}'_{\mathrm{T}})$ and vice versa.*

Proof Suppose a new vertex with the label A^0_j is added. Let us consider all the vertices in X to which this vertex is adjacent in $\mathcal{G}'_{\mathrm{T}}$. Then one of those vertices v must be in an equivalence class Y that had a vertex v' with an edge to vertex with label A^0_j in $\mathcal{G}_{\mathrm{T}}$. If there is no such Y, it implies that all the vertices in X adjacent to this new vertex had the label A^1_j in their label sequence before anonymization. Therefore, the new vertex with the label A^0_j can be removed and all equivalence classes of X will remain k-anonymous. Now note that v has both A^0_j and A^1_j in its label sequence in $\mathcal{G}'_{\mathrm{T}}$. Hence, the anonymization procedure must create a new vertex labelled A^1_j and add an edge between v' and this new vertex to preserve anonymity.

We now give a proof of correctness of our reduction.

Lemma 7 T *can be k-anonymized by suppressing t attributes if and only if the subset X of vertices in $\mathcal{G}_{\mathrm{T}}$ can be k-anonymized by adding $2t$ new vertices.*

Proof (only if:) Suppose T can be k-anonymized by suppressing t attributes. For every column A_j that is suppressed, we add two new vertices A_j^0 and A_j^1 and add an edge from every row vertex labelled R to one of these new vertices that was not in its label sequence originally. At the end of this procedure, if a set of rows of T formed a k-anonymous group, the corresponding set of row vertices in X form a k-anonymous group. To do this, we added $2t$ new vertices.
(if:) As shown in Lemma 6, we can assume that the anonymization procedure adds new vertices in pairs. If the procedure adds a new pair (A_j^0, A_j^1), then suppress the column A_j in T. Note that if $2t$ new vertices are added in $\mathcal{G}'_{\mathrm{T}}$, t attributes are suppressed in T. At the end of this procedure, it is easy to see that if a set of row vertices in X form a k-anonymous group in $\mathcal{G}'_{\mathrm{T}}$, then the corresponding set of rows of T are k-anonymous.

We now remark that this problem is also in *NP*. At the first glance, this is not clear as there is no natural bound on the integer t in terms of the size of the graph $\mathcal{G}$. However, based on Lemmas 3 and 4, it is clear that we can assume without loss of generality $t \leq |\Sigma|$. Therefore, a membership in this language can be shown by a list of at most $|\Sigma|$ new vertices and a list of new edges. Therefore, this problem is *NP*-Complete. This completes the proof of the theorem.

As a last note on the subject of tractibility, we remark that our algorithm given earlier implies tractibility (in fact, describes an $\mathcal{O}(nk)$ algorithm) for the analogous subset anonymization problem on unlabelled graphs. In the case of subset anonymization, the new vertices need not be also anonymized. The problem is interesting from a practical perspective, because it is non-obvious for an adversary to distinguish between vertices of V' and of $\mathrm{V} \setminus X$.

Corollary 2 *k-Labelled Subgraph Anonymization is in P if $|\Sigma| = 1$, for all k.*

Proof This follows from the algorithm in §3, because anonymization of a labelled subgraph with an alphabet size of 1 is equivalent to anonymization of an unlabelled graph simply by removing all the labels, performing the unlabelled anonymization, and then relabelling every vertex. Because $\mathrm{V}' \cap X = \emptyset$, the third step of the algorithm presented in §3 is not required. Consequently, exactly $\max \mathrm{deficiency}$ new vertices are sufficient to anonymize X, because the additional vertices were required for the anonymization of V'.

6 Related Work

The notion of k-anonymity was introduced by Sweeney [36] within the context of relational databases, with the insight that tabular microdata could be published without compromising privacy if every tuple of a relation was made to look identical to at least $k-1$ other tuples with respect to identifying and quasi-identifying attributes. Subsequent to this, work by Meyerson and Williams [32] and by Agarwal et al. [2] demonstrated hardness for k-anonymity of tables, even if the quasi-identifiers come from a small-sized alphabet. Outside the relational model, the privacy of statistical tables was studied towards inference control [14, 21, 35].

Later, the desire to publish microdata for– and, consequently the concern of privacy with respect to– other types of data emerged, particularly for the growing body of social network graphs. The knowledge transfer from the relational database community to the study of social networks was pioneered by Backstrom et al. [4], Zheleva and Getoor [42], and Hay et al. [22], who formalised the first notions of attacks against social network data.

This sparked a series of works on developing algorithms for anonymity against progressively stronger adversaries, originating with the study of Liu and Terzi [29] on degree-based attacks in unlabelled graphs and of Zhou and Pei [43] on neighbourhood attacks in vertex-labelled graphs. Both of these papers offered an experimentally effective algorithm, while the latter also offered a proof of *NP*-Hardness. The empirical merits of this approach was verified by Ying et al. [40], who demonstrated that deliberate k-anonymization can preserve structural characteristics of graphs much better than adding random noise does. Subsequently, Thompson and Yao [37] studied i-hop degree-based attacks. That is, an adversary's prior knowledge includes the degree of the target and the degree of its neighbors within i hops. They develop an inter-cluster matching method for anonymizing graphs against 1-hop attacks through edge addition and deletion. Cheng et al. [8], in their work on k-isomorphism, form k pairwise isomorphic subgraphs to achieve protection against two specific classes of attacks. Wu et al. [39] proposed the k-symmetry model, wherein for any vertex v, there exist at least $k-1$ other vertices to which v can be mapped using an automorphism of the underlying graph.

Each of the above adversarial models assumed that an adversary's objective was so-called *identity disclosure*: the identification of a vertex. Another attack is *attribute disclosure*, which seeks not necessarily to identify a vertex, but to reveal sensitive labels of the vertex. In the context of relational databases, this sort of attack was formalised by Machanavajjhala et al. [30] with the introduction of l-diversity as an anonymity measure, which requires that at least l attribute values appear in each equivalence class. Li et al. [28] then introduced t-closeness, which required instead that the *distribution* of attribute values in each equivalence class was within t of the entire dataset. More recently, Chester and Srivastava adopted these ideas for social networks with the introduction of α-proximity [12].

The need for these progressively stronger adversarial models stems from the difficulty in deriving an analagous notion for equivalence of tuples when adopting the k-anonymization model from relational databases. Despite all these results, our three questions from the introduction of hardness, approximation guarantees, and class-specific analysis are unaddressed for the majority of these models. Our work here, and, indeed, our introduction of a vertex-addition approach, is meant to deepen understanding of the known adversarial models in the same manner that the work of Agarwal et al. [2] and of Meyerson and Williams [32] did in the relational database setting.

Subset anonymity is a relatively new consideration. Studies by Yuan et al. [41] and by Ferri et al. [19] reveal that privacy concerns and needs are varied across and among different user groups. This has prompted a couple recent works [9, 11] that consider the more general k-subset-anonymity.

Another set of related works is as follows. König [24] showed that, given a graph $\mathcal{G}$ with maximum degree d, it is always possible to convert $\mathcal{G}$ into a d-regular graph H such that $\mathcal{G}$ is an induced subgraph of H. In a subsequent paper, Erdős and Kelly [16] strengthened the result of König by giving an efficient algorithm to determine the minimum number of new vertices that must be added to $\mathcal{G}$ to obtain such a graph H. This results were extended by the more recent the work of Akiyama et al. [3] and Bodlaender et al. [6]. This result can be viewed as an optimal n-degree-anonymization of $\mathcal{G}$. Our approach in this paper can be viewed as a generalisation of these results with two relaxations. First, as discussed by Yuan et al. [41], we may require that only a subset of nodes be anonymized. For example, different people have different expectations of privacy and some may not be concerned if they can be reidentified in the anonymous graph. Second, we require k-anonymity of the graph for an arbitrary k (which we typically assume to be some reasonably small value much less than d).

Our work here and the above references on k-anonymization, encompass at a high level "data anonymization" methods. These methods first transform the data and then release them. We note lastly that there is another general family of methods for achieving data privacy, which does not release data, but, rather, 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. The differentially-private methods (e.g., [15, 31]) belong in this family, which has a much different objective than anonymization.

7 Conclusion and Future Work

In this paper, we focused on the problem of k-degree-anonymizing a subset of a graph $\mathcal{G}$ to protect against a degree-based attack. We established theoretical results that hold for any input graph by introducing a natural alternative to the formulation of the problem present in literature. Specifically, we introduced a vertex-addition approach wherein, given $\mathcal{G} = (\mathrm{V}, \mathrm{E}, \Sigma, \ell)$, the new graph $\mathcal{G}' = (\mathrm{V} \cup \mathrm{V}', \mathrm{E} \cup \mathrm{E}', \Sigma \cup \Sigma', \ell \cup \ell')$ is constructed such that $\mathrm{E}' \subseteq \mathrm{V}' \times (\mathrm{V}' \cup \mathrm{V})$. The optimisation constraint is to minimize $|\mathrm{V}'|$, which ensures that $\mathcal{G}'$ is near to $\mathcal{G}$.

Within this setting, we established that k-degree-anonymizing a given subset X of V for $k \geq 3$ is *NP*-Complete, unless $|\Sigma| = 1$. For $|\Sigma| = 1$, we gave an efficient, exact algorithm for subset anonymization of X and an efficient, near-optimal algorithm to *also* ensure the k-degree-anonymity of all $\mathrm{X} \cup \mathrm{V}'$. Existing techniques in literature provide no characterisation of tractibility nor approximation guarantees.

Furthermore, we demonstrated empirically that the resultant graph $\mathcal{G}'$ of our algorithm is quite near to the input graph $\mathcal{G}$ with respect to five structural characteristics of graphs. We contrasted the output graphs for our algorithm with those produced by the state-of-the-art algorithm which does not allow a vertex set expansion, achieving highly comparable performance. This suggests that the vertex-addition approach performs well in real-life in addition to the strong theoretical claims that we have proven.

The success here implies that there are interesting research directions with which to extend this work. We focused on foundational work with a simpler adversarial model to establish the credibility of adding “fake” vertices. Clearly, extending the algorithmic work to more challenging adversaries would be well worthwhile since structural graph characteristics that may equally well preserved by doing so. Our hardness result dictates that an appropriate pursuit for the case of vertex-labeled graphs would be to design approximate or heuristic algorithsms. Finally, another research direction would be to enhance the algorithm by adding edges among $V' \times V'$ during the third step of our algorithm in a utility-oriented approach such as that introduced by Wang et al. [38].

References

1. Adamic, L., Glance, N.: The political blogosphere and the 2004 u.s. election: Divided they blog. In: Proc. WWW 2005 Workshop on the Weblogging Ecosystem (2005)
2. Aggarwal, G., Feder, T., Kenthapadi, K., Motwani, R., Panigrahy, R., Thomas, D., Zhu, A.: Anonymizing tables. In: Proc. International Conference on Database Theory (ICDT), pp. 246–258 (2005)
3. Akiyama, J., Era, H., Harary, F.: Regular graphs containing a given graph. In: American Mathematics Monthly v83, pp. 15–17 (1983)
4. Backstrom, L., Dwork, C., Kleinberg, J.M.: Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In: Proc. Conference on World Wide Web (WWW), pp. 181–190 (2007)
5. Barrat, A., Weigt, M.: On the properties of small-world network models. European Physics Journal B **13** (2000)
6. Bodlaender, H.L., Tan, R.B., van Leeuwen, J.: Finding a delta-regular supergraph of minimum order (2000)
7. Chakrabarti, D., Faloutsos, C.: Graph mining: Laws, generators, and algorithms. ACM Computing Surveys **38**(1), 2 (2006). DOI http://doi.acm.org/http://doi.acm.org/10.1145/1132952.1132954
8. Cheng, J., Fu, A.W.C., Liu, J.: K-isomorphism: privacy preserving network publication against structural attacks. In: Proc. ACM Special Interest Group on Management of Data (SIGMOD), pp. 459–470 (2010)
9. Chester, S., Gaertner, J., Stege, U., Venkatesh, S.: Anonymizing subsets of social networks with degree constrained subgraphs. In: Proc. Advances in Social Networks Analysis and Mining (ASONAM) (2012)
10. Chester, S., Kapron, B., Ramesh, G., Srivastava, G., Thomo, A., Venkatesh, S.: k-anonymization of social networks by vertex addition. In: Proc. Advances in Databases and Information Systems (ADBIS) (2011)
11. Chester, S., Kapron, B., Srivastava, G., Venkatesh, S.: Complexity of social network anonymization. Soc. Netw. Anal. Min. (2012)
12. Chester, S., Srivastava, G.: Social network privacy for attribute disclosure attacks. In: Proc. Advances in Social Networks Analysis and Mining (ASONAM) (2011)
13. Costa, L.d.F., Rodrigues, F.A., Travieso, G., Villas Boas, P.R.: Characterization of complex networks: A survey of measurements. Advances in Physics **56**, 167–242 (2007)
14. Domingo-Ferrer, J. (ed.): Inference Control in Statistical Databases, From Theory to Practice, *Lecture Notes in Computer Science*, vol. 2316. Springer (2002)
15. Dwork, C.: Differential privacy. In: in ICALP, pp. 1–12. Springer (2006)
16. Erdős, P., Kelly, P.: The minimal regular graph containing a given graph. In: American Mathematics Monthly v70, pp. 1074–1075 (1967)
17. Estrada, E., Rodriguez-Velazquez, J.A.: Spectral measures of bipartivity in complex networks. Physical Review E **72** (2005)
18. Faloutsos, M., Faloutsos, P., Faloutsos, C.: On power-law relationships of the internet topology. SIGCOMM Comput. Commun. Rev. **29**(4), 251–262 (1999). DOI 10.1145/316194.316229
19. Ferri, F., Grifoni, P., Guzzo, T.: New forms of social and professional digital relationships: the case of facebook. Soc. Netw. Anal. Min. **2**(2), 121–137 (2012)
20. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. In: Proc. Natl. Acad. Sci., vol. 99, pp. 7821–7826 (2002)

21. González, J.J.S.: Extending cell suppression to protect tabular data against several attackers. In: Inference Control in Statistical Databases, pp. 34–58 (2002)
22. Hay, M., Miklau, G., Jensen, D., Towsley, D.F., Weis, P.: Resisting structural re-identification in anonymized social networks. Proc. Very Large Databases (PVLDB) **1**(1), 102–114 (2008)
23. Heer, J.: Prefuse: a toolkit for interactive information visualization. In: In CHI 05: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 421–430. ACM Press (2005)
24. König, D.: Akademische verlagsgesellschaft. In: Leipzig (1936)
25. Latora, V., Marchiori, M.: Efficient behavior of small-world networks. Physics Review Letters **87** (2001). DOI 10.1103/PhysRevLett.87.198701
26. Leskovec, J., Kleinberg, J., Faloutsos, C.: Graphs over time: Densification laws, shrinking diameters and possible explanations. In: Proc. International Conference on Knowledge Discovery and Data Mining (KDD) (2005)
27. Leskovec, J., Lang, K.J., Dasgupta, A., Mahoney, M.W.: Statistical properties of community structure in large social and information networks. In: Proc. Conference on World Wide Web (WWW), pp. 695–704 (2008)
28. Li, N., Li, T., Venkatasubramanian, S.: t-closeness: Privacy beyond k-anonymity and l-diversity. In: Proc. of IEEE 23rd Intl Conf. on Data Engineering (ICDE07 (2007)
29. Liu, K., Terzi, E.: Towards identity anonymization on graphs. In: Proc. ACM Special Interest Group on Management of Data (SIGMOD), pp. 93–106 (2008)
30. Machanavajjhala, A., Kifer, D., Gehrke, J., Venkitasubramaniam, M.: L-diversity: Privacy beyond k-anonymity. ACM Trans. Knowl. Discov. Data **1**(1) (2007). DOI 10.1145/1217299.1217302
31. McSherry, F., Mironov, I.: Differentially private recommender systems: Building privacy into the netflix prize contenders. In: Proc. International Conference on Knowledge Discovery and Data Mining (KDD), pp. 627–636 (2009)
32. Meyerson, A., Williams, R.: General k-anonymization is hard. In: Principles of Database Systems (2004)
33. Milgram, S.: The small world problem. In: Psychology Today, vol. 2, pp. 60–67 (1967)
34. Newman, M.E.J.: Finding community structure in networks using the eigenvectors of matrices (2006)
35. Robertson, D.A., Ethier, R.: Cell suppression: Experience and theory. In: Inference Control in Statistical Databases, pp. 8–20 (2002)
36. Sweeney, L.: k-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **10**(5), 557–570 (2002)
37. Thompson, B., Yao, D.: The union-split algorithm and cluster-based anonymization of social networks. In: Proc. ACM Symposium on Information, Computer and Communications Security (ASIACCS), pp. 218–227 (2009)
38. Wang, Y., Xie, L., Zheng, B., Lee, K.C.K.: Utility-oriented k-anonymization on social networks. In: Proceedings of the 16th international conference on Database systems for advanced applications - Volume Part I, DASFAA'11, pp. 78–92. Springer-Verlag, Berlin, Heidelberg (2011)
39. Wu, W., Xiao, Y., Wang, W., He, Z., Wang, Z.: k-symmetry model for identity anonymization in social networks. In: Proc. International Conference on Extending Database Technology (EDBT), pp. 111–122 (2010)
40. Ying, X., Pan, K., Wu, X., Guo, L.: Comparisons of randomization and k-degree anonymization schemes for privacy preserving social network publishing. In: Proc. 3rd Workshop on Social Network Mining and Analysis (SNA-KDD), pp. 10:1–10:10. ACM, New York, NY, USA (2009)
41. Yuan, M., Chen, L., Yu, P.S.: Personalized privacy protection in social networks. Proceedings of Very Large Databases, PVLDB **4**(2), 141–150 (2010)
42. Zheleva, E., Getoor, L.: Preserving the privacy of sensitive relationships in graph data. In: Proc. Privacy, Security, and Trust in KDD (PinKDD), pp. 153–171 (2007)
43. Zhou, B., Pei, J.: The *k* -anonymity and *l* -diversity approaches for privacy preservation in social networks against neighborhood attacks. Knowledge Information Systems **28**(1), 47–77 (2011)