True Friends Let You Down: Benchmarking Social Graph Anonymization Schemes

Kumar Sharad
University of Cambridge
kumar.sharad@cl.cam.ac.uk

ABSTRACT

Greater demand for social graph data among researchers and analysts has fueled an increase in such datasets being published. Consequently, concerns about privacy breach have also risen steadily. To mitigate privacy risks a myriad of social graph anonymization schemes have been proposed. Anonymizing high dimensional data is a very hard problem and conventionally it is considered unwise to publish graph data even without identifiers. Often the schemes proposed provide no proof of efficacy and are designed to defeat only a narrow set of attacks.

To facilitate benchmarking of perturbation-based social graph anonymization schemes we propose a machine learning framework which provides a quick and automated platform to evaluate and compare the schemes. We present a mechanism to train the framework without ground truth. We present graph structure based node features that can be easily tuned to accommodate weak or strong adversaries as well as node and edge attributes. The framework provides a granular graph structure-based metric to capture the likelihood of a node being re-identified. We conduct a thorough analysis of the effect of graph perturbation on anonymity achieved and utility preserved using publicly available real world social graphs. To this end we analyze six popular graph perturbation schemes including those promising k-anonymity. Our techniques automate weeding out poor anonymization schemes. Experiments show that it is hard to provide anonymity while preserving utility whereas some schemes destroy utility without providing much anonymity. All *useful* anonymization schemes leave a fraction a true edges intact and these *true* friends lead to the re-identification of nodes.

1. INTRODUCTION

Much of the data collected as a result of our digital activities is high dimensional and a valuable resource to study human behavior. As a result the demand for such data among researchers is very high, to fill this gap data providers often release data for public good. However, this poses serious threat to the privacy of individuals in the dataset [1–3]. High dimensional data is notorious to anonymize [4], even more so for social graphs due to entities being interlinked.

Social graphs provide a rich source of data for studying human behavior. However, such benefits come at a price; releasing feature rich private datasets have a massive potential to cause a privacy catastrophe[1]. To alleviate these concerns data providers often scrub off personal identifiers claiming it to be sufficient to preserve privacy. Such techniques are easily defeated [1–3]. Though it provides the two fold advantage of simplicity and legal compliance by providing plausible deniability in case of a privacy breach.

Considerable effort has been spent in devising better social graph anonymization schemes that preserve privacy without hindering analysis [5–9], however without carrying the burden of proof. It should be noted that some amount of information would inevitably be destroyed to preserve privacy thus affecting analysis. There is an inherent tension between preserving privacy vs preserving utility of datasets and one cannot be achieved without adversely affecting the other. Most schemes are proposed in an ad-hoc manner and show no incremental evolution, thus confounding their comparison. This has resulted in numerous anonymization schemes, but constructing attacks even for simple ones requires careful study and manual work. This has created a skewed ecosystem where anonymization schemes are proposed without much effort and a considerable time must be spent to evaluate them.

As a solution, we propose a machine learning framework to benchmark perturbation-based social graph anonymization schemes. To the best of our knowledge this is the first attempt to compare anonymization schemes in a quick and automated manner. The framework can efficiently handle any perturbation-based social graph anonymization scheme and a wide variety of threat models.

Our contributions. Specifically, we: (*i*) Design and implement a machine learning framework to benchmark perturbation-based social graph anonymization schemes. The framework provides a quick and automated platform to evaluate and compare schemes efficiently (Section 3.3). (*ii*) Present a mechanism to train the framework without ground truth (Section 3.3.2). (*iii*) Conduct a thorough analysis of the effect of graph perturbation on the anonymity achieved and utility preserved using publicly available real world social graphs. To this end we analyze six popular graph perturbation schemes including those promising k-anonymity based on two real world social networks for three perturbation levels each – a total of 36 configurations (Section 4). (*iv*) Conduct a thorough analysis of the effect of graph perturbation on utility by analyzing five fundamental graph metrics for each of the 36 configurations (Section 4).

2. ANONYMIZING SOCIAL GRAPHS

Social graph anonymization schemes mainly fall under two categories [10–12] – clustering-based schemes and perturbation-based schemes.

AISec'16, October 28 2016, Vienna, Austria

ISBN 978-1-4503-4573-6/16/10. . . $15.00
DOI: http://dx.doi.org/10.1145/2996758.2996765

[1] www.nytimes.com/2006/08/09/technology/09aol.html

2.1 Clustering-based Schemes

Clustering-based graph anonymization schemes release aggregate graph information instead of the raw graph. Zheleva and Getoor [13] propose preserving privacy in social graphs by removing sensitive edges and selectively deleting non-sensitive edges. Cormode *et al.* [14] propose perturbing the mapping from entity to nodes using safe groupings while keeping the structure of the graph intact. Hay *et al.* [15] propose preserving privacy by grouping nodes into partitions and publishing the number of nodes in each partition and density of edges that exist within and across partitions. Campan and Truta [16] present a clustering mechanism similar to Hay *et al.*, the authors also consider preserving privacy when the nodes have attributes. Bhagat *et al.* [17] propose using label lists to protect the privacy of node attributes and partitioning nodes into classes to protect against structural attacks. Clustering nodes and edges provides some privacy with limited data utility. Summarizing information prevents granular analysis and only provides an aggregate view. Perturbation-based schemes fare better in terms of data utility which we discuss next.

2.2 Perturbation-based Schemes

Perturbation-based schemes introduce imperfections in the social graph before publishing to deter graph-structure-based de-anonymization attacks. Certain schemes appear frequently in literature, they all involve deleting/adding edges in various ways. Some of them are: (*i*) Random Sparsification (`RSP`) [2, 5] – delete a fraction of graph edges at random. (*ii*) Random Add/Delete (`RAD`) [5, 18, 19] – delete a number of edges followed by introducing the same number of non-edges at random, this preserves the total number of edges in the graph. (*iii*) Random Switch (`RSW`) [5, 19–21] – randomly select two edges and switch them across their nodes which are not already connected. This preserves the number of edges and the individual node degrees.

Ying and Wu [19] modify `RAD` and `RSW` to preserve spectral graph properties which are often damaged by random perturbation. Liu *et al.* [22] propose preserving edge weight privacy of a released graph. Xue *et al.*(Random Edge Perturbation – `REP`) [8] propose perturbing the graph by removing a fraction of edges and adding the same fraction of non-edges to defeat structure based attacks [23].

k-Anonymity Based Schemes. Latanya Sweeney [24] proposed k-anonymity to anonymize relational databases. A released dataset is k-anonymous if each individual is indistinguishable from at least $k - 1$ others in the dataset. k-anonymity provides some desirable properties; however, it is of little use in protecting high-dimensional datasets [4]. Despite this a multitude of k-anonymity based graph anonymization schemes have been proposed. Such schemes show a gradual progression – starting with making the degrees of nodes k-anonymous followed by 1-hop neighborhood to 2-hop neighborhood (see, Appendix A). The schemes that target neighborhoods aim to modify the sub-graph around each node in such a way that it becomes indistinguishable from $k - 1$ others. Such schemes are computationally very expensive and some are known to be NP-hard [6, 9]. Also they are particularly damaging to the overall utility of graph; the nature of anonymization strategy makes analysis of community structure and related properties hard.

Liu and Terzi (k-Degree Anonymous – `KDA`) [7] propose ways to make the graph k-degree anonymous. Zhou and Pei (1-hop k-Anonymous – `1HKA`) [25] present a scheme to make the nodes in a social network k-anonymous with respect to it's 1-hop neighborhood. Thompson and Yao [26] propose schemes similar to `KDA` and `1HKA` to achieve k-degree anonymity and 1-hop k-anonymity using node clustering. Zou *et al.* [9] propose an anonymization scheme based on graph isomorphism. The original graph is modified by inserting nodes and edges such that each node has at least $k - 1$ automorphisms. Cheng *et al.* [6] propose techniques to prevent leakage of a node's identifying information and its relationship with other nodes in anonymized social networks. They propose a solution using k-isomorphism by splitting the original graph into k isomorphic disjoint sub-graphs such that they are pair-wise isomorphic. Wu *et al.* [27] propose k-symmetry to obscure node identities by inserting new edges and vertices to perturb a graph. The graph is anonymized by forcing each node to have at least $k - 1$ structurally equivalent counterparts to prevent structural attacks.

As k-anonymization guarantees get stronger the complexity rises exponentially and we run into NP-hard problems. Such schemes also require suppressing huge amounts of information [4] thus rendering the data useless. Remainder of the paper analyses a representative sample of six anonymization schemes – `RSP`, `RAD`, `RSW`, `REP`, `KDA` and `1HKA` in detail and compares them with the base case when graphs are not anonymized. The analysis studies the true marginal anonymity achieved purely due to the anonymization scheme employed and its effect on utility.

3. QUANTIFYING ANONYMITY

Quantifying anonymity in graphs is hard—even for a given anonymization scheme it is challenging to quantify the relation between anonymity and graph perturbation. It is much harder to compare different schemes and the anonymity they provide. All meaningful anonymization schemes are constrained by preserving utility which makes them vulnerable to attacks. We exploit the utility constraint to train a learning algorithm that learns features from anonymized graphs to quantify de-anonymization success.

3.1 The Threat Model

Research indicates that anonymizing high-dimensional data to preserve privacy does not work in practice [1–3]. Even when an anonymized dataset alone does not pose any threat of privacy breach, when combined with auxiliary data they could lead to discoveries which were not possible in the absence of the anonymized data.

Releasing anonymized social graphs presents similar challenges. Even after removing node identifiers the structure of the graph can be used to splice it with overlapping social graphs thus revealing scrubbed off identities and potentially private relations among them. Ideally an anonymization scheme should render relinking attacks [2, 3] impractical while still retaining the utility of datasets. Graph anonymization literature contains a wide variety of schemes with varying aims (see, Section 2). Comparing these schemes requires a common threat model. To measure the efficacy of an anonymization scheme we quantify the success rate in re-identifying common nodes in an overlapping pair of graphs, both of which have been anonymized using the scheme being analyzed. The adversary uses one of the graphs as background knowledge to attack the other graph. This is similar to the setting used by Narayanan and Shmatikov [2]. One of the other popular threat models [5, 18, 19] assumes that adversary knows the target degree, this is unrealistic but easy to evaluate hence popular. We cannot consider a weaker threat model because attacks [2] already exist under the current threat model. We focus on structure based attacks as most of the graph perturbation based schemes are designed to conceal the graph structure and prevent re-identification based on node neighborhood. However, due to its modularity the threat model and learning model can incorporate a variety of adversaries which are much more advanced. We describe them in greater detail below.

3.2 Graph Generation

Benchmarking an anonymity scheme begins with generating a pair of graphs with an intersecting node set from real world graphs [2]. Each graph is anonymized using the scheme to be benchmarked. We treat one graph as the target graph (mimicking the sanitized version released) and the other graph as the auxiliary graph at attacker's disposal (side information for linking identities), these can be interchanged. We constrain the attacker to only have the knowledge of graph structure. This is the least amount of information which is released and inclusion of any other information such as node or edge attributes only strengthens the attacker.

We take a real world social network $G = (V, E)$ and randomly partition the set of nodes V into two subsets V_1 and V_2 with an overlap α_V (measured by Jaccard Coefficient; see, Appendix A). An overlap of α_V is obtained by randomly partitioning V into three subsets V_A, V_B and V_C with sizes $\frac{1-\alpha_V}{2} \cdot |V|$, $\alpha_V \cdot |V|$ and $\frac{1-\alpha_V}{2} \cdot |V|$ respectively and setting $V_1 = V_A \cup V_B$ and $V_2 = V_B \cup V_C$, in our experiments we use $\alpha_V = 0.25$. Finally, we create two graphs G_1 and G_2 as node induced sub-graphs of G using vertex sets V_1 and V_2. G_1 and G_2 are subsequently anonymized using the scheme to be benchmarked, the anonymized auxiliary and sanitized graphs thus produced are called G_{aux} and G_{san} respectively. We refer to their vertex sets as V_{aux} and V_{san} respectively, note that $V_{aux} = V_1$ and $V_{san} = V_2$. The nodes and edges are stripped of all identifiers, we only consider the structure of the graphs in further discussions. This allows us to quantify the efficacy of the anonymization scheme in preserving privacy by measuring the success of structure-based re-identification. The efficacy is measured in terms of success of the classifier in differentiating whether two individuals belonging to social graphs G_{aux} and G_{san} are identical or not.

3.3 The Classification Framework

We propose a machine learning based classification framework that uses structure-based re-identification to measure the privacy leak in social networks. The classifier can quantify anonymity of social graph anonymization schemes; the framework is based on an ensemble of randomly trained decision trees known as random decision forest [28]. The forest is trained to classify node pairs (n_{aux}, n_{san}) such that $n_{aux} \in V_{aux}$ and $n_{san} \in V_{san}$ as identical or non-identical using the features of each node.

3.3.1 Node Features

Node similarity metrics have been well known to significantly improve the confidence and accuracy in predicting links between nodes [21, 29]. Degree distribution is a fundamental property of social networks [30] and a node can be uniquely identified by its neighborhood degree distribution [18]. Our framework uses the degree distribution of 1-hop and 2-hop neighbors as features to describe a node. These are generic graph features which are not tied to any anonymization scheme and perform well for a variety of learning tasks [3, 31]. Both 1-hop and 2-hop features are computed separately and then concatenated. Each component in the feature vector quantizes the number of nodes with degree in a given range, as shown in Figure 1, $c_0 = 8$ means there are 8 1-hop nodes with degrees in the range (0-50), $c_1 = 4$ implies 4 1-hop nodes with degrees in range (51-100) and so on. In our experiments we chose the vector length to be 21 and bin size to be 50, all degrees exceeding the maximum range are included in the last bin.

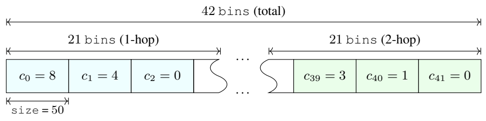

Figure 1: Example node feature vector

We also use the Silhouette Coefficient (see, Appendix A) of degrees of a node pair as a feature. The modularity of features allows us to tune them according to the adversary, e.g. for a directed graph the features could be expanded to contain the in and out degree of 1-hop and 2-hop neighborhoods instead of combining the degrees.

3.3.2 Training

One of our key contributions is presenting a mechanism to train a model in the absence of ground truth or seed mappings across auxiliary and sanitized data. In the scenario of a graph release an adversary does not possess the real data but a damaged version of it. The adversary attempts to splice it with the data at their disposal to re-identify individuals. Training a machine learning model is tricky in such a scenario as we need access to the ground truth. To re-identify nodes with high confidence it is important to train the model with high quality data. Ideally, the model should be trained by generating auxiliary and sanitized graphs from the same graph that was used to generate G_{aux} and G_{san}. Providing the adversary access to such data makes it too strong and unrealistic. We circumvent this by training the classifier using node pairs from G_{aux} and G_{san}, both the graphs are split again to produce two sets of overlapping graphs. This time however, we do not apply any anonymization on the generated graphs. This allows us to sample node pairs from the overlapping sets for which we know the ground truth. Data is sampled from each set and merged to train the classifier. Only a rough estimate of the overlap between auxiliary and sanitized graphs is sufficient to sample data thus producing node pairs which closely resemble those being attacked.

We also experimented by sampling training data by generating auxiliary and sanitized graph from a different but similar social network to that under attack. The learning task is transferable [3, 31] thus cross training works as well but we get better results by training from a distribution which is as close to the original as possible. Splitting G_{aux} and G_{san} to train simulates training under ideal circumstances as closely as possible. G_{aux} and G_{san} represent a damaged version of the original graph (G_1 and G_2 respectively), however, under our threat model they are the closest dataset to that being attacked and hence we only need to split them as they have already undergone perturbation. Training the forest allows it to learn features that optimize the classification success. The trees are trained by randomly sampling training data and node features; we use a forest of 400 trees.

3.3.3 Classification

After training the decision forest a pair of feature vectors (v_{aux}, v_{san}) representing the node pair (n_{aux}, n_{san}) sampled from G_{aux} and G_{san} is passed through the forest. Each tree assigns a probability to the pair of being identical or non-identical. After the node pair has passed through all the trees we average the predictions to reach a final prediction.

4. EVALUATION AND RESULTS: ANONYMITY vs. UTILITY

We benchmark graph anonymization schemes based on quality of anonymization and utility preservation. For each of the six social graph anonymization schemes – RSP, RAD, RSW, REP, KDA and 1HKA, we measure how de-anonymization success and utility vary versus strength of anonymization. Intuitively, if an increase in anonymization does not produce a commensurate decrease in de-anonymization success while substantially diminishing utility then such an anonymization scheme is considered less favorably. All the schemes being evaluated provide varying levels of *anonymity* by controlling the level of graph perturbation. We note that increasing perturbation does not necessarily provide more anonymity in all cases but it always affects utility adversely.

The schemes are evaluated using two publicly available real world social graphs – Flickr (nodes = 80 513, edges = 5 899 882) [32] and Facebook New Orleans dataset (nodes = 63 731, edges = 817 090) [33]. The Flickr graphs generated for benchmarking (see, Section 3.2) have about 50 000 nodes and 2 310 000 edges while graphs for Facebook have about 40 000 nodes and 320 000 edges prior to any anonymization.

Interpreting the ROC curves. For a given social graph anonymization scheme and anonymity strength, each node pair passed through the classifier is assigned a score in [0, 1]. This procedure is carried out for more than a million randomly selected node pairs to analyze the success of structure-based re-identification. The score assigned to each node pair signifies its likelihood of being non-identical. An ideal classifier will output 1 whenever it sees a non-identical node pair and a 0 whenever it sees an identical node pair. The Receiver Operating Characteristic (ROC) curves illustrate how close the classifier is to an ideal one. It does so by measuring the True Positive (TP) rate as the False Positive (FP) rate tolerated is varied in the range [0, 1]. An ideal classifier gives a TP rate of 1 at FP rate 0, whereas TP and FP are always the same for random guessing. In practice a classifier will always make errors (FP), our goal is to maximize the correct classification rate (TP) for the error tolerated. The Area Under the Curve (AUC) provides a summary of the quality of the classifier, an ideal classifier has an AUC = 1 where as random guessing produces a classifier with an AUC = 0.5.

Measuring Anonymity. Anonymity of a scheme is measured by the de-anonymization success achieved as depicted by ROC curves and the AUC (figure legend); this allows us to compare schemes. We also compare the performance of the classifier when the graph is simply split (denoted as GS) and no anonymization is applied (node pairs are sampled from G_1 and G_2) to the case where a particular scheme is used (node pairs are sampled from G_{aux} and G_{san}).

Measuring Utility. Measuring utility is harder as there is no standard metric to capture it and is usage dependent. An anonymization scheme might perfectly preserve the degree distribution of a graph while damaging other properties. We look at some fundamental utility metrics as they vary with anonymization level: (*i*) Degree distribution (DD) – it measures the frequency of degrees as they grow and is an important measure of a small world graph. (*ii*) Joint degree distribution (JDD) – the distribution of node degree pairs between which edges exist. (*iii*) Average degree connectivity (ADC) – the average nearest neighbor degree of nodes with degree k. (*iv*) Degree centrality (DC) – the fraction of total nodes a node is connected to. (*v*) Eigenvector centrality (EVC) – it measures a node's importance based on its connection to other important nodes. For a vertex v it is defined as the v^{th} component of the eigenvector associated with the largest eigenvalue of the adjacency matrix of the graph.

These properties are fundamental to social graphs and most real-world utility metrics are derived from them. Significant damage to these metrics adversely affects utility. Figure 2 shows the JDD of unperturbed graphs for reference.

Implementation. The project has been implemented in Python and run using CPython. We use a commodity laptop with 2.8 GHz processor and 16 GB RAM for our experiments. The classifier is trained using about 25K identical and 500K non-identical node pairs. We classify 16K identical node pairs for Flickr and 10K for Facebook against 1M non-identical node pairs for both graphs. To focus on a typical social network user, only nodes with degree over five are studied. The number of identical node pairs is lower for Facebook as it is sparser. Training the classifier takes about 10 minutes while classification takes about 25 minutes.

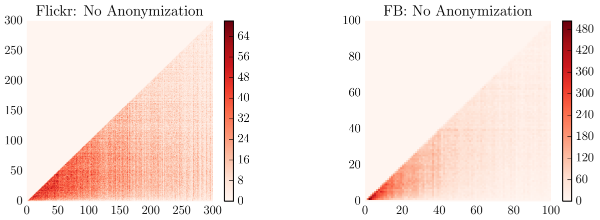

Figure 2: Original Joint Degree Distribution

4.1 Random Sparsification (RSP)

Anonymity. Deleting edges at random limits the scope of structural de-anonymization due to lack of information. As shown in Figure 3 the classification success diminishes with decreasing edge overlap (as measured by Jaccard Coefficient). We introduce an edge overlap of $\alpha_E = (0.75, 0.50, 0.25)$ by increasing the fraction of randomly deleted edges to produce G_{aux} and G_{san}; α_E is computed for the common subgraph of G_{aux} and G_{san}, the edge overlap for the entire graph is much lower. Even after lowering α_E to 0.25, enough information remains and de-anonymization is quite successful.

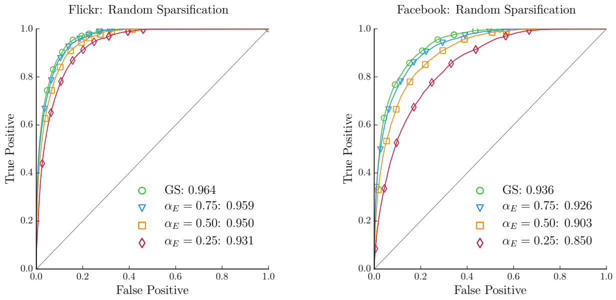

Figure 3: RSP: ROC curves

Utility. Deleting edges shows gradual and graceful decline in the quality of the graph. (*i*) DD (Figure 4) – more of less preserved for varying levels of anonymity, this is expected as edges are deleted uniformly at random. (*ii*) JDD (Figure 5) – shifts towards the low degree node pairs as deleting edges decreases the number of high degree nodes. (*iii*) ADC (Figure 6) – shifts towards the origin due to decrease in node degrees across the graph. Decrease in the range of degrees shrinks the spectrum. (*iv*) EVC (Figure 7) – not much affected, the important nodes continue to be important. However, the decrease of degree does shift the spectrum.

4.2 Random Add/Delete (RAD)

Anonymity. RAD homogenizes the graph by decreasing the degree of high degree nodes and increasing the degree of low degree nodes as non-edges are more likely exist between such nodes. As a result structural classification becomes difficult (Figure 8). We compare the performance of our classifier by introducing graph perturbation of a fraction $k = (0.10, 0.25, 0.50)$ of the edges. Even at $k = 0.50$ de-anonymization is quite successful for both the graphs.

Utility. RAD is less forgiving of the graph properties as the graph is pushed closer to randomness (achieved at $k = 1$). (*i*) DD (Figure 9) – addition and deletion of edges at random makes the DD more compact. The shift is disproportionate due to nature of perturbation be-

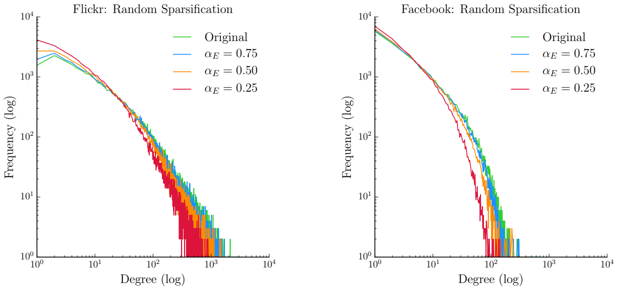

Figure 4: RSP: Degree Distribution

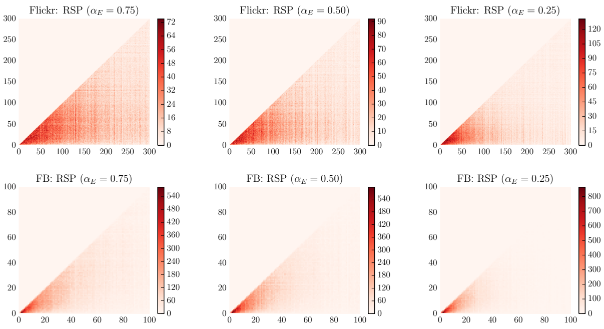

Figure 5: RSP: Joint Degree Distribution

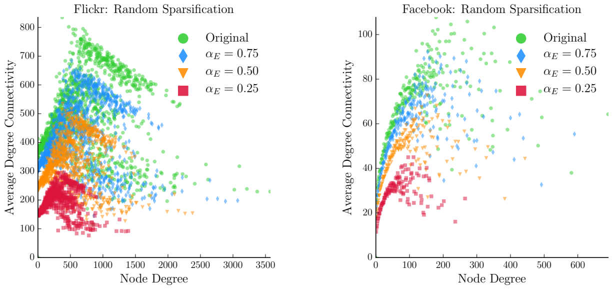

Figure 6: RSP: Degree Connectivity vs. Node Degree

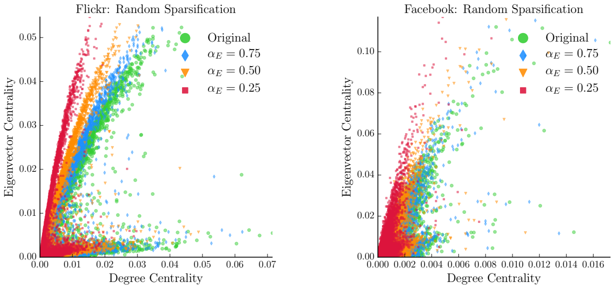

Figure 7: RSP: EVC vs. DC

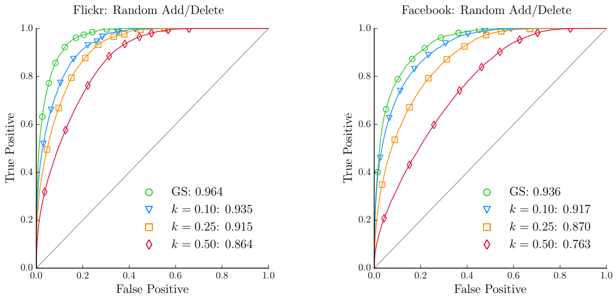

Figure 8: RAD: ROC curves

ing biased. (*ii*) JDD (Figure 10) – confirms that all the node degrees move close together. (*iii*) ADC (Figure 11) – shifts downwards as low degree nodes get delinked from high degree nodes and linked to other low degree nodes, the high degree nodes suffer a decrease in their degree and loss of links to other high degree nodes, this decreases their connectivity as well. Since node degrees become uniform the spectrum shrinks. (*iv*) EVC (Figure 12) – perturbation of the neighborhood of high degree nodes decreases their degree; however, they still retain their importance as deleting and adding edges at random does not have a huge effect on their influence. On the other hand low degree nodes still remain relatively unimportant because of being primarily connected to other low degree nodes thus producing the shift observed.

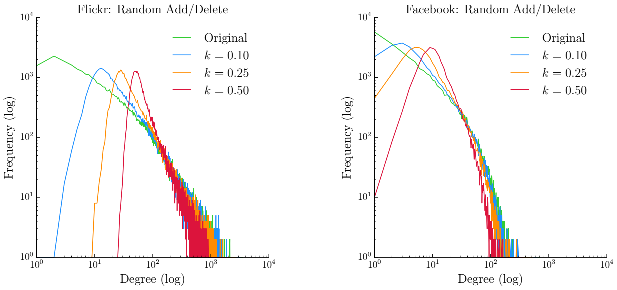

Figure 9: RAD: Degree Distribution

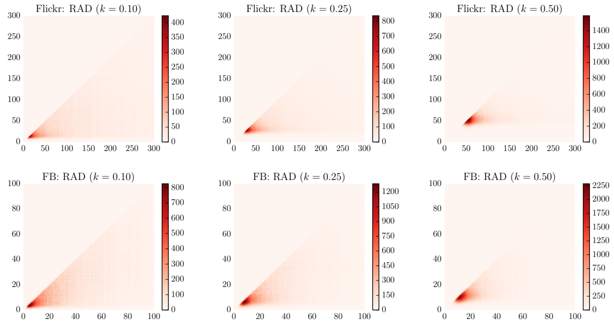

Figure 10: RAD: Joint Degree Distribution

4.3 Random Switch (RSW)

Anonymity. RSW effects the graph properties in an unpredictable manner. We introduce perturbations of a fraction $k = (0.20, 0.50, 0.85)$ of the number of edge pairs; smaller values of k produced no perceptible change in ROC curves hence larger values are picked to study variance. Figure 13 shows that even at the highest level of perturbation barely any additional privacy is achieved. Preserving the DD not only adversely effects graph's properties but also makes it more vulnerable to structure-based re-identification.

Utility. Although DD is perfectly preserved, other graph metrics are

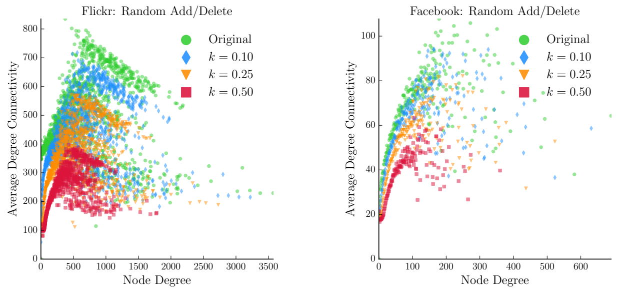

Figure 11: RAD: Degree Connectivity vs. Node Degree

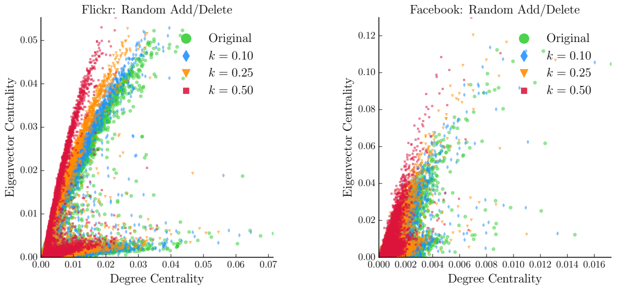

Figure 12: RAD: EVC vs. DC

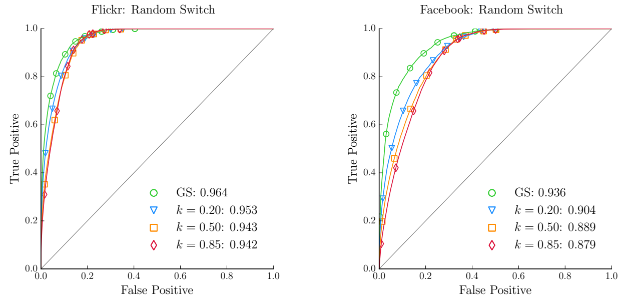

Figure 13: RSW: ROC curves

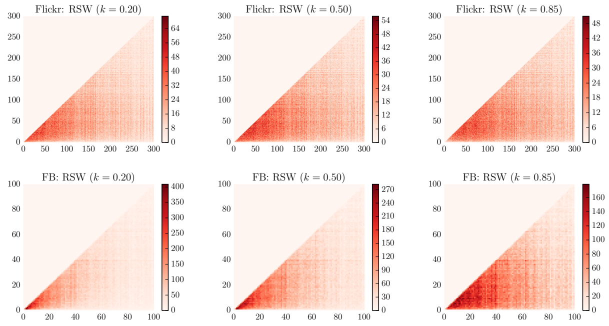

Figure 14: RSW: Joint Degree Distribution

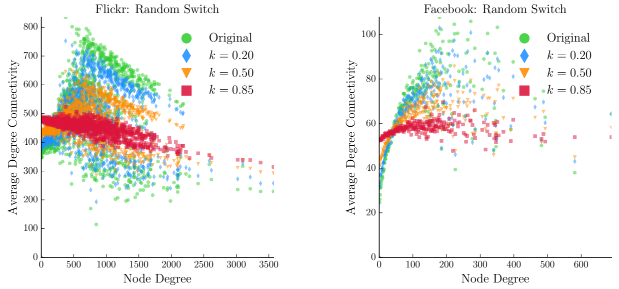

Figure 15: RSW: Degree Connectivity vs. Node Degree

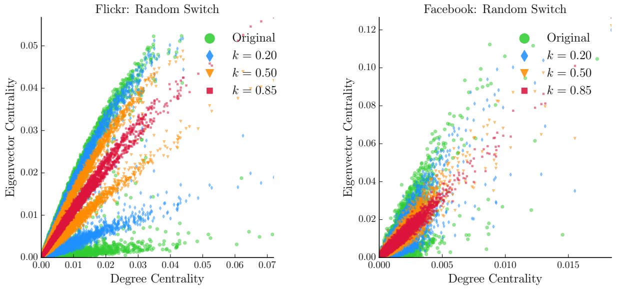

Figure 16: RSW: EVC vs. DC

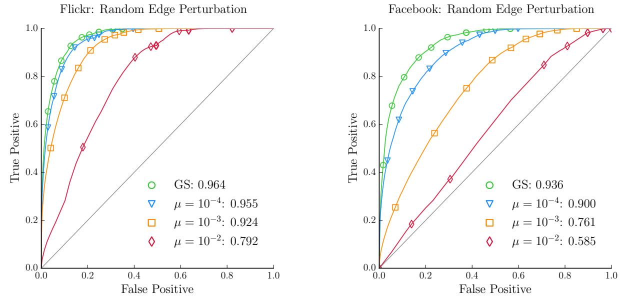

Figure 17: REP: ROC curves

profoundly damaged thus highlighting the challenges faced in preserving utility in a perturbed graph. (*i*) DD – exactly preserved. (*ii*) JDD (Figure 14) – becomes more uniform throughout the graph. The original JDD (Figure 2) shows that edges are concentrated among low degree nodes. Switching edges gradually spreads the concentration towards higher degree nodes. (*iii*) ADC (Figure 15) – becomes uniform as low degree and high degree nodes get linked. (*iv*) EVC (Figure 16) – remains preserved for low levels of perturbation ($k = 0.10$) but the influence of nodes becomes directly proportional to DC at highest perturbation. This is an indication of the graph losing structure and moving towards randomness.

4.4 Random Edge Perturbation (REP)

Anonymity. Deleting a fraction of edges and adding the same fraction of non-edges produces a large increase in edges overall since the number of non-edges is several orders of magnitude higher than edges. We introduce perturbations of $\mu = (10^{-4}, 10^{-3}, 10^{-2})$. Figure 17 shows that denser graphs are more resilient to noise; this is even more apparent in the case of REP which is rather damaging to Facebook at $\mu = 10^{-3}$ but does little damage to Flickr.

Utility. Although we achieve a reasonable level of anonymity at $\mu = 10^{-3}$ for Facebook it comes at the cost of utility which is significantly damaged. Flickr resists longer but to achieve anonymity we need to compromise utility. At $\mu = 10^{-2}$ both graphs are perturbed beyond recognition and of little use but even at this level of perturbation Flickr still looks attackable with AUC = 0.792, where as an attack on Facebook is same as guessing with AUC = 0.585.

(*i*) DD (Figure 18) – shifts towards the right but the change is more extreme as the proportion of non-edges introduced is orders of magnitude higher. (*ii*) JDD (Figure 19) – gets concentrated towards the high degree node pairs. (*iii*) ADC (Figure 20) – addition of edges produces new low degree nodes that are connected to other low degree nodes thus producing a dip of connectivity spectrum towards the origin. The neighborhood of high degree nodes shows relatively less change. (*iv*) EVC (Figure 21) – in contrast to `RSP` adding non-edges at random does not change the DC or EVC of nodes much. Random edges do not change the importance of nodes, hence the spectrum is preserved. However, extreme perturbation ($\mu = 10^{-2}$) causes the spectrum to shift.

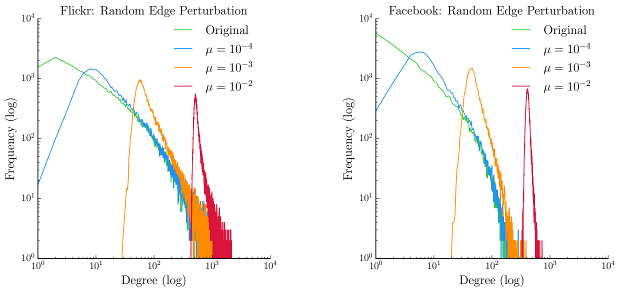

Figure 18: REP: Degree Distribution

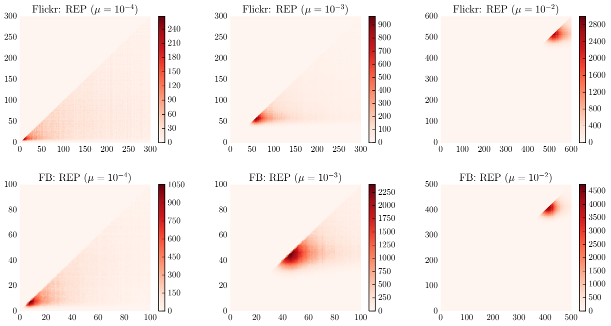

Figure 19: REP: Joint Degree Distribution

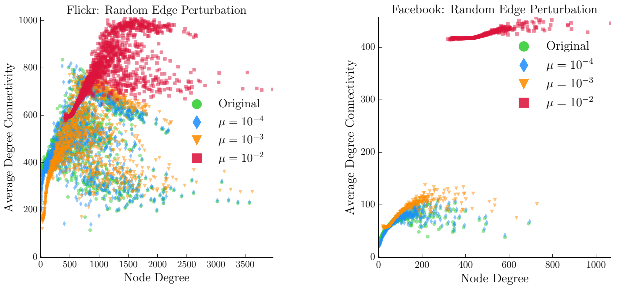

Figure 20: REP: Degree Connectivity vs. Node Degree

4.5 k-Degree Anonymization (`KDA`)

Anonymity. `KDA` introduces edges among high degree nodes which are rarer. We anonymize graphs [7] using `Supergraph` which produces a k-anonymous graph for a given number of nodes followed by `Greedy_Swap` which swaps edges among node pairs till the generated graph has almost the same edge set as the original graph. In our experiments the generated graph has at least 90% of the same edges as the original graph. Perturbations of $k = (10, 50, 100)$ are introduced for analysis. As observed (Figure 22) even high values of k does not increase the anonymity by much for either graph. Edge insertion among high degree nodes is not large enough to mask their true neighborhood structure whereas the low degree nodes are left almost untouched. The result is significant damage to graph metrics without gaining much anonymity.

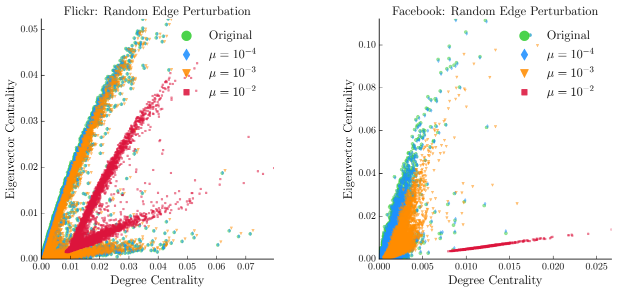

Figure 21: REP: EVC vs. DC

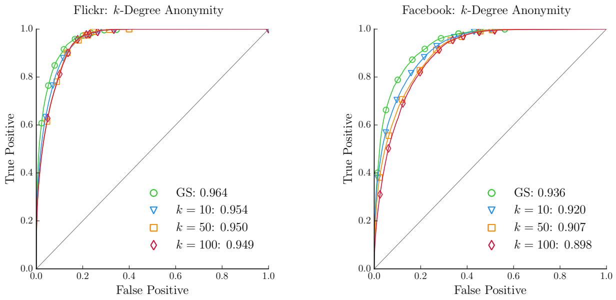

Figure 22: KDA: ROC curves

Utility. Change in the neighborhood of high degree nodes adversely affects the properties of graph without purchasing any additional anonymity. (*i*) DD (Figure 23) – flatlines for higher degree nodes as they are fewer in number and the perturbation increases the frequency of each distinct degree to a minimum value. (*ii*) JDD (Figure 24) – forms a checkered pattern for high degree nodes due to introduction of new edges among high degree nodes. (*iii*) ADC (Figure 25) – increases sharply for low degree nodes due to being connected to high degree nodes whose degree has been increased. (*iv*) EVC (Figure 26) – remains largely same except vertical patterns appear for high degree nodes (higher DC) due to perturbation.

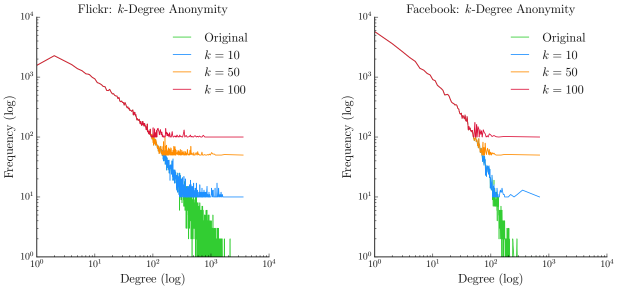

Figure 23: KDA: Degree Distribution

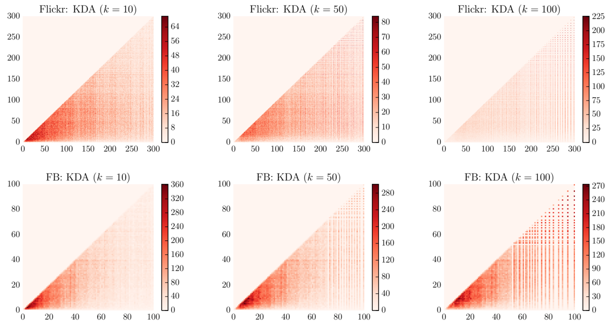

Figure 24: KDA: Joint Degree Distribution

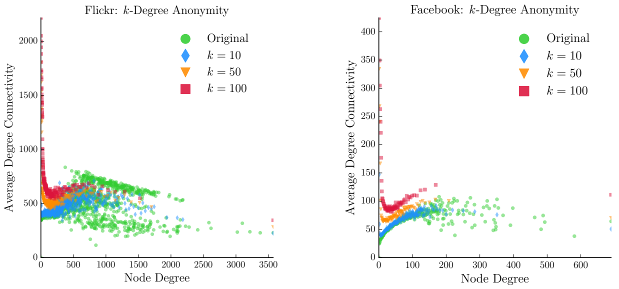

Figure 25: KDA: Degree Connectivity vs. Node Degree

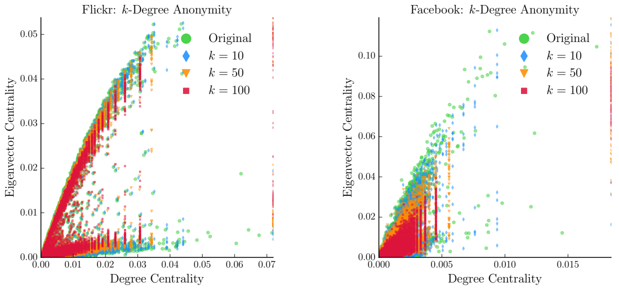

Figure 26: KDA: EVC vs. DC

4.6 1-hop k-Anonymization (1HKA)

Anonymity. 1HKA is inefficient for large graphs with high average node degree. The scheme ensures 1-hop k-anonymity by inserting edges which are in the order of 12% of the total edges for $k = 30$ [25]. To analyze this scheme and circumvent the problem of inefficiency we introduce $\mu = (0.10, 0.25, 0.50)$ fraction of edges at random in the graph. Inserting edges at random makes structural de-anonymization harder as compared to inserting them in a formulaic manner (as for KDA), also utility is better preserved in this scenario. Hence the results achieved provide a lower bound on de-anonymizability and an upper bound on utility compared to using the actual scheme. Note that by definition 1-hop features of all nodes would be completely identical in this scenario therefore they cannot be used to differentiate among node pairs any more. The 2-hop features are still relevant; we replace the 1-hop features by 3-hop features. This is a simple swap which is easily done in our framework thus highlighting it's proclivity to swift adaptation. Figure 27 confirms that even adding a high fraction of edges does not provide much anonymity.

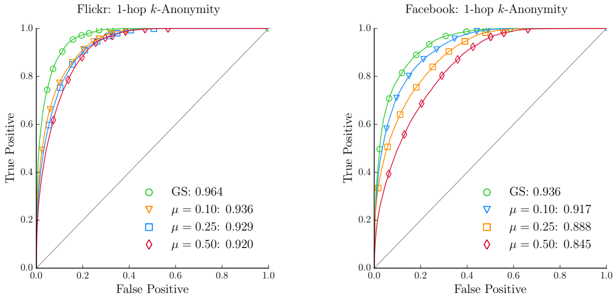

Figure 27: 1HKA: ROC curves

Utility. Random edge addition damages DD and JDD but is gentler on other properties. (*i*) DD (Figure 28) – similar to REP. (*ii*) JDD (Figure 29) – similar to REP, shifts diagonally as all node degrees increases. The shift is concentrated towards the low degree nodes as most edges are introduced in their vicinity. (*iii*) ADC (Figure 30) – similar to REP. (*iv*) EVC (Figure 31) – similar to REP.

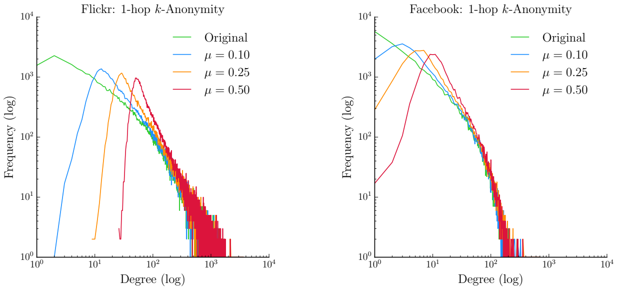

Figure 28: 1HKA: Degree Distribution

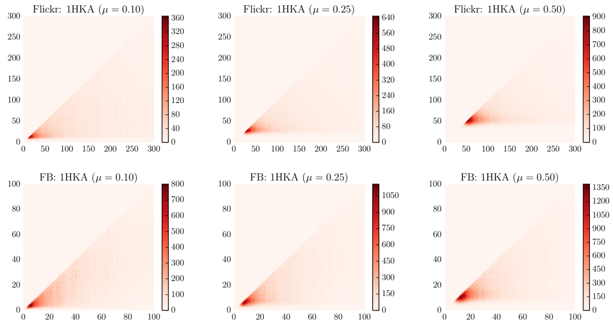

Figure 29: 1HKA: Joint Degree Distribution

5. RELATED WORK

A-posteriori analysis of re-identification probabilities is a popular [5, 18, 19] approach to quantify anonymity in anonymized social graphs. Given a threat model such analysis formulates queries based on the adversary's a-priori knowledge of the target and then computes the risk of re-identification based on the induced equivalence relation. Such a methodology is hard to quantify due to being computationally intensive. The results are highly dependent on the threat model and can only handle simple adversaries [5]. Constructing optimal queries for an adversary possessing knowledge of the target's sub-graph is very tricky.

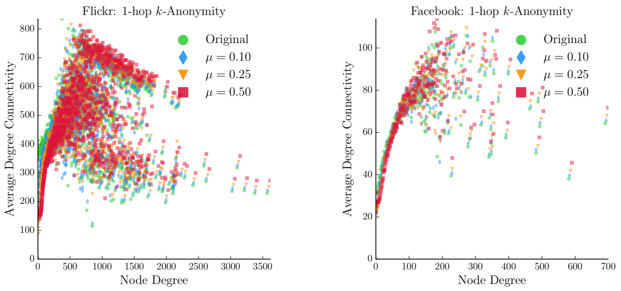

Figure 30: 1HKA: Degree Connectivity vs. Node Degree

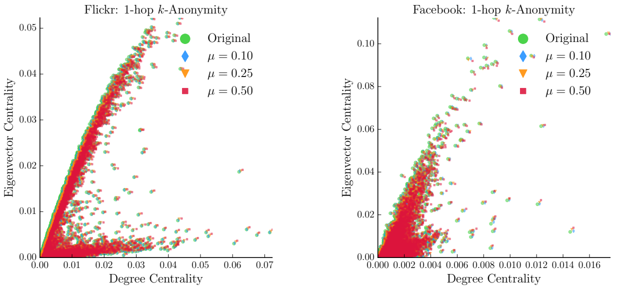

Figure 31: 1HKA: EVC vs. DC

Hay *et al.* [18] present an adversary that knows some structural information about a target in the original graph and tries to identify it by querying the published anonymized graph. All the resulting candidates of a query are considered equally likely and each query is enforced to have at least k candidates. The authors also present an adversary who explores the neighborhood of the target by performing a breadth first search. This models the scenario where the adversary has incomplete knowledge of the sub-graph around the target. Both the adversaries are unrealistic as they do not look for a probabilistic match or optimize for more informative edges and the k-anonymity is based on node degree.

Bonchi *et al.* [5] refine the previous approach by proposing an entropy based metric. Hay *et al.*measure the probability of a node in the perturbed graph to have originated from the target, thus providing a local estimate. Bonchi *et al.*provide a global estimate by computing the entropy from the distribution of belief probabilities of the nodes being mapped to the target. The authors model an adversary with knowledge of target's degree and compute entropy for a graph anonymized using RSP. These computations however, become far too complicated and computationally infeasible for even a slightly more potent adversary that has the knowledge of target neighborhood instead of just the target degree.

Ji *et al.* [34] conduct a survey of graph anonymization schemes based on utility preservation and their resistance to de-anonymization attacks. The threat model used is impractical as it does not capture a plausible scenario. Using pure graph de-anonymization attacks to compare graph anonymization schemes is not generalizable; for instance if an attack uses information which is not present e.g. edge directionality [2]. If the attack uses seeds then the results vary greatly based on the seed quality. Additionally, attacks can be optimized to de-anonymize a large fraction of nodes while sacrificing accuracy or vice-versa, they can also be tuned to be more successful for certain nodes than others. Graph de-anonymization attacks are not designed to serve as a measuring tool. Our framework can easily adapt to changes in threat model and accommodate a weak or strong adversary. This is in stark contrast to the rigidity of structure-based de-anonymization attacks which are time consuming to modify and cannot be upgraded or degraded without manual effort. None of the attacks presented are successful when the anonymized graph is published as a collection of disjoint subgraphs. Availability of seeds would be useless as the mapping would stop at the limit of each subgraph, our framework can handle such cases with ease [3]. Most approaches are computationally intensive (e.g. Bonchi *et al.* [5]) and thus heavy footed. We provide a nimble and automated alternative to benchmark social graph anonymization schemes.

Sharad and Danezis [3] were the first to use machine learning techniques to de-anonymize social networks. Our work goes much beyond the scope of previous work and bridges the fundamental research gap of comparing and evaluating social graph anonymization schemes under a common framework.

6. DISCUSSION

Anonymizing high dimensional data is not a new problem, though it keeps resurfacing in different forms. Data with a large number of attributes are very hard to anonymize without unacceptable information loss [4]. There are exponentially many combinations of dimensions that can be used as quasi-identifiers to mount inference attacks. Preventing such attacks requires completely suppressing most of the data which renders its publication moot. Our results confirm the conventional wisdom in the scenario of social graphs.

Parameter choice. The overlap between G_{aux} and G_{san} is set as $\alpha_V = 0.25$ to model an adversary with reasonable side information to mount an attack. A higher α_V strengthens the attacker as the side information increases; however, it does not impact the *relative* success of attacks under different schemes and threat models. We set vector length and bin size (n, b) as $(21, 50)$ (see Section 3.3.1); the value is chosen to accommodate higher degrees and their variation, the choice does not impact accuracy [3]. Testing accuracy increases monotonically with the forest size [28, 35]; Criminisi *et al.* [28] obtain good results with forest size of 400, which is what we use. We experimented by using features such as centrality, edge weights and group membership in addition to those proposed. Complicated features do not provide significant improvement over those used.

6.1 Risk of Re-identification

The classification framework presented quantifies the risk of re-identification based purely on structure given a perturbation scheme. The task of distinguishing identical node pairs from non-identical ones captures the most basic challenge faced by the adversary. We provide a granular graph structure-based metric to capture the likelihood of a node being re-identified. The classification task used by our framework is not new and has been widely used in the literature [1, 2, 36]. The key difference is that previous results have been reported for the global matching task rather than pairwise matching task. The results of the global matching task are derived from the success attained in pairwise matching task. Hay *et al.* [18] proposed *K-Candidate Anonymity* based on the number of matches returned by a structural query on a graph. Bonchi *et al.* [5] refine the definition to propose k*-Preimage Obfuscation* which is based on an entropy. Both these definitions rely on the success of the adversary to find pairwise matchings of graph nodes.

Our structure-based re-identification has a high true positive rate and a low false positive rate, leading to re-identification of an individual with high confidence. The re-identification rates observed are a lower bound on the attacker's success. Existence of more potent structure-based re-identification cannot be ruled out. Additionally, structure is not the only information that an adversary might possess, any side information is likely to cause further decline in false positives. Hence, we find that none of the six schemes analyzed provide sufficient anonymity while preserving utility (see, Appendix B Figure 32). Acceptable levels of anonymity are only achieved at very high perturbation levels at which point data is of little use. REP ($\mu = 10^{-2}$) for Facebook is the most successful scheme at repelling structural attacks with true positive value of less than 1% for a false positive of 0.1% and an AUC = 0.585; though not as effective for Flickr it behaves reasonably well. We introduce extreme levels of perturbation in our experiments only as a means to study the graph behavior, perturbation at such high levels serves no practical purpose.

Flickr. All the schemes except REP ($\mu = 10^{-2}$) have a TP of above 5% at a FP of 0.1% for even the most extreme level of perturbation. This is already pretty high even if we make the very strong assumption that the attacker can gather no side information for any of the nodes attacked.

Facebook. Anonymization is more successful as compared to Flickr. Most schemes apart from REP ($\mu = (10^{-2}, 10^{-3})$) have a TP of around 3–5% at a FP of 0.1%. This also effects the utility of graph metrics which are reduced. Overall, Facebook's anonymization and utility are more sensitive given a particular level of perturbation.

6.2 Scheme Comparison

Bearing in mind that none of the schemes analyzed here are really fit for the purpose, we provide a coarse ranking based on utility preservation and relative anonymity. RSW and KDA are by far the least useful schemes for anonymizing graphs. Neither provide a safe level of anonymity even after hugely damaging the graph features. RSP and RAD are by far the best schemes among the ones that we have analyzed; both provide graceful and gradual degradation of graph utility. REP and 1HKA lie in the middle. Schemes that provide k-anonymity by homogenizing the n-hop neighborhood around nodes are computationally expensive and defenseless against subgraph attacks. Such schemes are very destructive to graph properties and can be defeated by extending the features beyond n-hops.

Appendix B, Table 1 provides the concrete relation between the deviation of DD and JDD from the original as characterized by the Hellinger distance (see, Appendix A) and AUC of the ROC curves. The schemes that show a lesser distance between original and perturbed distributions preserve the distribution better. In general schemes which produce distributions closer to the original ones tend to allow more successful structural attacks, which supports our analysis. Appendix B, Figure 32 shows a comparison of the schemes for a chosen anonymization strength.

Bottom line. After careful analysis we do not believe any scheme can guarantee anonymity while preserving utility. De-anonymization can be seen as a utility metric as it is constructed from graph properties and it cannot be damaged in isolation. In summary: (*i*) Our experiments show that properties of dense graphs are more resilient to a proportionate perturbation which in turn makes them more vulnerable to attacks. De-anoymization of Flickr is less sensitive to edge perturbations than Facebook as dense graphs retain sufficient information even after perturbation. Additionally, high degree nodes are more vulnerable than low degree nodes. All useful anonymization schemes leave a fraction of true edges intact and only a few *true* friends are needed to re-identify an individual. Thus dense graphs are more vulnerable to re-identification attacks as compared to sparse graphs since on an average each node has more friends. (*ii*) Deleting edges is less harmful than adding false edges [5]. Introducing random edges disrupts the small world characteristics of the graph by shrinking it, while removing edges at random still leaves paths that preserve the small world features. (*iii*) Increasing perturbation does not necessarily provide more anonymity in all cases but it always degrades utility. (*iv*) Formulaic and local graph perturbation such as KDA fares worse at providing anonymity than global graph perturbation. The interconnectivity of graphs allows leveraging the unperturbed neighborhoods to attack the perturbed neighborhoods. An anonymization scheme must be global to have any chance of providing privacy. (*v*) Discovery of more potent structure-based re-identification would not alter the *relative* ranking of the schemes. Structure-based re-identification which is generalizable would always be more successful on a weaker scheme. The classification framework presented is generic as its success increases with decrease in strength of anonymization for a particular scheme. This claim is supported by the results presented for six different schemes.

7. CONCLUSION

It has always been easy to propose graph anonymization schemes, but hard to assess whether they actually work. We provide a framework that levels the playing field for the first time by automating the analysis of such schemes. Quick and automated analysis empowers the data holders to swiftly triage newly proposed schemes. We show how to train a classifier in the absence of ground truth by generating subgraphs and sampling data from auxiliary and sanitized graphs. The classifier uses node features that can adapt to changes in threat model (as demonstrated for 1HKA) and accommodate adversaries of varying strength. Incremental features allows us to model adversaries with much more sophisticated queries based on neighborhood and can be easily modified to include node and edge attributes. Traditional structure-based graph attacks cannot readily adapt to changing adversaries. Additionally, using attacks to compare schemes is not ideal as vulnerability to a particular attack does not capture the true extent of a scheme's failure since near misses are not considered when producing full mapping. Measuring true positive versus false positive rate is far more granular and gives us more information. Unlike attack based measurement our framework does not assume a model of the adversary which can tolerate only a certain amount of error and hence rigid. We perform a detailed analysis of six perturbation-based social graph anonymization schemes. A thorough study of trade-off between anonymity and utility as a function of graph perturbation is also presented.

Our threat model considers an adversary which has access to imperfect structural information of the graph which is used to identify members of intersecting graphs. Both the graph at the adversary's disposal and the released graph are aggressively and synthetically damaged which limits the adversary. In practice it is highly likely that the adversary can get hold of a graph that has been damaged as a result of organic processes which are not adversarial. In such a scenario the attacks will be more potent and catastrophic for the privacy of individuals whose data is released.

We believe taking a conservative approach while releasing data is the best way forward. Even though anonymization schemes are useful and should be employed to safeguard privacy they are not a solution. Such schemes should be used to dissuade the curious but honest adversary but do little to stop a malicious adversary if utility preservation is important. Social graph anonymization schemes and anonymization schemes dealing with high dimensional data in general should always be backed up by legal agreements which prohibit malicious use of data.

Acknowledgments. We thank Laurent Simon, Ross Anderson and Steven Murdoch for their comments on the draft manuscripts. This work was supported by the Engineering and Physical Sciences Research Council [grant number EP/J500665/1]; and Microsoft Research through its PhD Scholarship Programme.

References

[1] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *2008 IEEE Symposium on Security and Privacy (S&P 2008), 18-21 May 2008, Oakland, California, USA*, pp. 111–125, 2008.

[2] A. Narayanan and V. Shmatikov, "De-anonymizing social networks," in *Proceedings of the 2009 30th IEEE Symposium on Security and Privacy*, SP '09, (Washington, DC, USA), pp. 173–187, IEEE Computer Society, 2009.

[3] K. Sharad and G. Danezis, "An automated social graph de-anonymization technique," in *Proceedings of the 13th Workshop on Privacy in the Electronic Society*, WPES '14, (New York, NY, USA), pp. 47–58, ACM, 2014.

[4] C. C. Aggarwal, "On k-anonymity and the curse of dimensionality," in *Proceedings of the 31st International Conference on Very Large Data Bases*, VLDB '05, pp. 901–909, VLDB Endowment, 2005.

[5] F. Bonchi, A. Gionis, and T. Tassa, "Identity obfuscation in graphs through the information theoretic lens," in *Proceedings of the 2011 IEEE 27th International Conference on Data Engineering*, ICDE '11, (Washington, DC, USA), pp. 924–935, IEEE Computer Society, 2011.

[6] J. Cheng, A. W.-c. Fu, and J. Liu, "K-isomorphism: Privacy preserving network publication against structural attacks," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, SIGMOD '10, (New York, NY, USA), pp. 459–470, ACM, 2010.

[7] K. Liu and E. Terzi, "Towards identity anonymization on graphs," in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, (New York, NY, USA), pp. 93–106, ACM, 2008.

[8] M. Xue, P. Karras, R. Chedy, P. Kalnis, and H. K. Pung, "Delineating social network data anonymization via random edge perturbation," in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, (New York, NY, USA), pp. 475–484, ACM, 2012.

[9] L. Zou, L. Chen, and M. T. Özsu, "K-automorphism: A general framework for privacy preserving network publication," *Proc. VLDB Endow.*, vol. 2, pp. 946–957, Aug. 2009.

[10] X. Wu, X. Ying, K. Liu, and L. Chen, "A survey of privacy-preservation of graphs and social networks," in *Managing and Mining Graph Data* (C. C. Aggarwal and H. Wang, eds.), vol. 40 of *Advances in Database Systems*, pp. 421–453, Springer US, 2010.

[11] E. Zheleva and L. Getoor, "Privacy in social networks: A survey," in *Social Network Data Analytics* (C. C. Aggarwal, ed.), pp. 277–306, Springer US, 2011.

[12] B. Zhou, J. Pei, and W. Luk, "A brief survey on anonymization techniques for privacy preserving publishing of social network data," *SIGKDD Explor. Newsl.*, vol. 10, pp. 12–22, Dec. 2008.

[13] E. Zheleva and L. Getoor, "Preserving the privacy of sensitive relationships in graph data," in *Proceedings of the 1st ACM SIGKDD International Conference on Privacy, Security, and Trust in KDD*, PinKDD'07, (Berlin, Heidelberg), pp. 153–171, Springer-Verlag, 2008.

[14] G. Cormode, D. Srivastava, T. Yu, and Q. Zhang, "Anonymizing bipartite graph data using safe groupings," *Proc. VLDB Endow.*, vol. 1, pp. 833–844, Aug. 2008.

[15] M. Hay, G. Miklau, D. Jensen, D. Towsley, and P. Weis, "Resisting structural re-identification in anonymized social networks," *Proc. VLDB Endow.*, vol. 1, pp. 102–114, Aug. 2008.

[16] A. Campan and T. M. Truta, "Data and structural k-anonymity in social networks," in *Privacy, Security, and Trust in KDD, Second ACM SIGKDD International Workshop, PinKDD 2008, Las Vegas, NV, USA, August 24, 2008, Revised Selected Papers*, pp. 33–54, 2008.

[17] S. Bhagat, G. Cormode, B. Krishnamurthy, and D. Srivastava, "Class-based graph anonymization for social network data," *Proc. VLDB Endow.*, vol. 2, pp. 766–777, Aug. 2009.

[18] M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava, "Anonymizing social networks," *Computer Science Department Faculty Publication Series*, p. 180, 2007.

[19] X. Ying and X. Wu, "Randomizing social networks: a spectrum preserving approach," in *Proceedings of the SIAM International Conference on Data Mining, SDM 2008, April 24-26, 2008, Atlanta, Georgia, USA*, pp. 739–750, 2008.

[20] X. Ying and X. Wu, "Graph generation with prescribed feature constraints," in *Proceedings of the SIAM International Conference on Data Mining, SDM 2009, April 30 - May 2, 2009, Sparks, Nevada, USA*, pp. 966–977, 2009.

[21] X. Ying and X. Wu, "On link privacy in randomizing social networks," in *Proceedings of the 13th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, PAKDD '09, (Berlin, Heidelberg), pp. 28–39, Springer-Verlag, 2009.

[22] L. Liu, J. Wang, J. Liu, and J. Zhang, "Privacy preserving in social networks against sensitive edge disclosure," tech. rep., Technical Report Technical Report CMIDA-HiPSCCS 006-08, Department of Computer Science, University of Kentucky, KY, 2008.

[23] L. Backstrom, C. Dwork, and J. M. Kleinberg, "Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography," in *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*, pp. 181–190, 2007.

[24] L. Sweeney, "K-anonymity: A model for protecting privacy," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, pp. 557–570, Oct. 2002.

[25] B. Zhou and J. Pei, "The k-anonymity and l-diversity approaches for privacy preservation in social networks against neighborhood attacks," *Knowledge and Information Systems*, vol. 28, no. 1, pp. 47–77, 2011.

[26] B. Thompson and D. Yao, "The union-split algorithm and cluster-based anonymization of social networks," in *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, ASIACCS '09, (New York, NY, USA), pp. 218–227, ACM, 2009.

[27] W. Wu, Y. Xiao, W. Wang, Z. He, and Z. Wang, "K-symmetry model for identity anonymization in social networks," in *Proceedings of the 13th International Conference on Extending Database Technology*, EDBT '10, (New York, NY, USA), pp. 111–122, ACM, 2010.

[28] A. Criminisi, J. Shotton, and E. Konukoglu, “Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning,” *Foundations and Trends in Computer Graphics and Vision*, vol. 7, no. 2-3, pp. 81–227, 2012.

[29] D. Liben-Nowell and J. Kleinberg, “The link prediction problem for social networks,” in *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, CIKM ’03, (New York, NY, USA), pp. 556–559, ACM, 2003.

[30] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, “Measurement and analysis of online social networks,” in *IMC ’07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, (New York, NY, USA), pp. 29–42, ACM, 2007.

[31] K. Henderson, B. Gallagher, L. Li, L. Akoglu, T. Eliassi-Rad, H. Tong, and C. Faloutsos, “It’s who you know: Graph mining using recursive structural features,” in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’11, (New York, NY, USA), pp. 663–671, ACM, 2011.

[32] R. Zafarani and H. Liu, “Social computing data repository at ASU,” 2009.

[33] B. Viswanath, A. Mislove, M. Cha, and P. K. Gummadi, “On the evolution of user interaction in facebook,” in *Proceedings of the 2nd ACM Workshop on Online Social Networks, WOSN 2009, Barcelona, Spain, August 17, 2009*, pp. 37–42, 2009.

[34] S. Ji, W. Li, P. Mittal, X. Hu, and R. Beyah, “Secgraph: A uniform and open-source evaluation system for graph data anonymization and de-anonymization,” in *24th USENIX Security Symposium (USENIX Security 15)*, (Washington, D.C.), pp. 303–318, USENIX Association, Aug. 2015.

[35] J. Shotton, M. Johnson, and R. Cipolla, “Semantic texton forests for image categorization and segmentation,” in *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*, 2008.

[36] G. Wondracek, T. Holz, E. Kirda, and C. Kruegel, “A practical attack to de-anonymize social network users,” in *31st IEEE Symposium on Security and Privacy, S&P 2010, 16-19 May 2010, Berleley/Oakland, California, USA*, pp. 223–238, 2010.

APPENDIX

A. DEFINITIONS

1. Ego – A graph node around which the egonet is formed.
2. Egonet – The local network centered around an ego derived based on some function.
3. n-hop node – A node such that the shortest path length from the node to the ego is n.
4. n-hop network – A node induced neighborhood graph around an ego with all n-hop nodes included. It is also known as n-hop neighborhood.
5. Jaccard Coefficient between sets X and Y at least one of which is non-empty is defined as: $JC(X,Y) = \frac{|X \cap Y|}{|X \cup Y|}$.
6. Silhouette Coefficient of degrees of two nodes belonging to G_{aux} and G_{san} is defined as: $\delta(d_1, d_2) = \frac{|d_1 - d_2|}{\max(d_1, d_2)}$, where $d_1 = \text{degree}(n_{aux})$, $n_{aux} \in V_{aux}$ and $d_2 = \text{degree}(n_{san})$, $n_{san} \in V_{san}$.
7. Hellinger Distance – The statistical distance $H(P,Q)$ between two discrete probability distributions $P = (p_1, \ldots, p_k)$ and $Q = (q_1, \ldots, q_k)$ is defined as: $H(P,Q) = \frac{1}{\sqrt{2}}\sqrt{\sum_{i=1}^{k}(\sqrt{p_i} - \sqrt{q_i})^2}$.

B. ADDITIONAL DETAILS

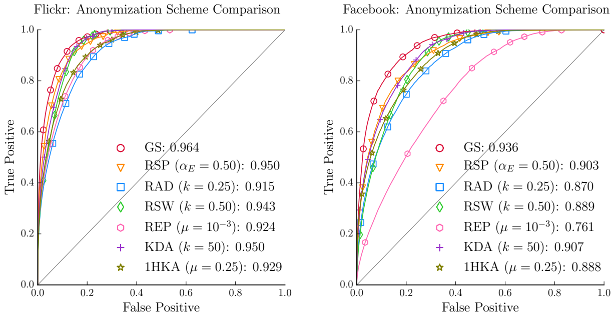

Figure 32: Scheme Comparison: ROC curves

Table 1: Hellinger Distance between DD and JDD for perturbed and unperturbed graphs and it’s effect on AUC

	Flickr			Facebook		
	DD	JDD	AUC	DD	JDD	AUC
RSP ($\alpha_E = 0.75$)	0.109	0.570	0.959	0.062	0.295	0.926
RSP ($\alpha_E = 0.50$)	0.130	0.567	0.950	0.100	0.340	0.903
RSP ($\alpha_E = 0.25$)	0.204	0.610	0.931	0.194	0.477	0.850
RAD ($k = 0.10$)	0.314	0.562	0.935	0.138	0.283	0.917
RAD ($k = 0.25$)	0.467	0.582	0.915	0.273	0.336	0.870
RAD ($k = 0.50$)	0.603	0.657	0.864	0.439	0.478	0.763
REP ($\mu = 10^{-4}$)	0.232	0.568	0.955	0.280	0.286	0.900
REP ($\mu = 10^{-3}$)	0.599	0.612	0.924	0.759	0.630	0.761
REP ($\mu = 10^{-2}$)	0.912	0.899	0.792	0.999	1.000	0.585
1HKA ($\mu = 0.10$)	0.318	0.562	0.936	0.141	0.269	0.917
1HKA ($\mu = 0.25$)	0.465	0.581	0.929	0.281	0.293	0.888
1HKA ($\mu = 0.50$)	0.584	0.606	0.920	0.428	0.338	0.845
RSW ($k = 0.20$)	0.000	0.170	0.953	0.000	0.084	0.904
RSW ($k = 0.50$)	0.000	0.268	0.943	0.000	0.143	0.889
RSW ($k = 0.85$)	0.000	0.350	0.942	0.000	0.203	0.879
KDA ($k = 10$)	0.136	0.576	0.954	0.056	0.279	0.920
KDA ($k = 50$)	0.260	0.793	0.950	0.125	0.485	0.907
KDA ($k = 100$)	0.327	0.863	0.949	0.175	0.603	0.898