Threatening Privacy across Social Graphs: A Structural Features Approach

Priya Govindan Tina Eliassi-Rad Jin Xu Shawndra Hill Chris Volinsky
Rutgers University BehaviorMatrix, LLC University of Pennsylvania AT&T Labs Research
{priyagn, eliassi}@cs.rutgers.edu jxu@behaviormatrix.com shawndra@wharton.upenn.edu volinsky@research.att.com

***Abstract*—Can privacy of individuals in an anonymized social graph be threatened by a handful of structural features−i.e., without knowledge of link-level connectivity? The answer is yes. We define *threatening privacy* as being able to narrow down each anonymized individual's identity to a small set of known individuals that is most likely to include the anonymized individual. We call this set the *Identity*$^+$ or I^+ set. In this manner, some anonymized individuals get associated with smaller I^+ sets and others to larger I^+ sets. This distinction is how their privacy is threatened. To find I^+ sets, we utilize (1) the structural features released in the anonymized graph (such as a node's degree, clustering coefficient, and average degree of neighbors) and (2) information in an auxiliary (publicly available) graph. To our knowledge, our work is the first of its kind that only uses structural features of graphs for deanonymization. Previous works assume the adjacency matrix of the anonymized graph is released and hence rely on the sparsity of the human behavior exhibited in the adjacency matrix. We will show that structural features are not sparse (i.e., have few zero entries). In fact, there are many lookalikes when one *only* inspects structural features. Our proposed approach, *RRID+*, recursively clusters and matches structural features of the anonymized and auxiliary graphs in a top-down greedy manner. Averaged over a range of real graphs, RRID+ is able to find the correct cluster in the auxiliary data for 37% of the individuals (i.e., recall = 37%) and narrow down their identity to 26% of the population in the auxiliary data (i.e., precision on recalled nodes = 74%). Our experiments−involving multiple synthetic and real graphs plus different noise models (between the anonymized and auxiliary graphs)−showcase how various parameters can affect precision and recall.**

I. Introduction

Imagine an institution (such as the National Institute of Health) wanting like to release social network data for research purposes. The identities must be anonymized, but simple label-anonymization has been shown to be potentially re-identifiable in the hands of an adversary [1]. As an alternative the institution decides to release only the structural features of each node in the social network−allowing researchers to study graph properties but avoiding the potential privacy breach. Re-identifying people with merely structural features is challenging since structural feature matrices (unlike adjacency matrices) are not sparse (i.e., they have very few zeros).

Can an adversary threaten the privacy of individuals in an anonymized social graph if only a handful of structural features (such as node degree, node clustering coefficient, average degree of a node's neighbors) are released? As we shall demonstrate, the answer is yes. Previous research [2] showed that 87% of the U.S. population are uniquely identified by date of birth, gender, and ZIP code. Other works [1] showed that a small partial matching on a graph's connectivity (e.g., a social network's adjacency matrix) can reveal identities. In those cases, only intrinsic features or connectivity of relationships are released. We study how an adversary can use structural features to discriminate between the anonymized individuals.

We define *threatening privacy* as a *relative* concept. Suppose an adversary is able to narrow down each anonymized individual's identity to the small set of known individuals that is most likely to include the anonymized individual.[1] We call this set the *Identity*$^+$ or I^+ set. In this manner, some anonymized individuals get associated with smaller I^+ sets and others to larger I^+ sets. This distinction is how their privacy is threatened−namely, individuals who get associated with smaller I^+ sets are more "distinguishable" than ones who get associated with larger I^+ sets. For instance, consider an anonymized co-authorship graph. Looking at the distributions of the released features, an adversary can figure out that the anonymized data is a social bibliographic graph (e.g., because it has many more connected components than a friendship graph). He/she can then use a publicly available bibliographic data to distinguish between the super-stars (i.e., individuals whose I^+ sets are small) and others (i.e., individuals whose I^+ sets are large.) Armed with this information, he/she can further discriminate between the super-stars by other means (e.g., label the I^+ sets based on the auxiliary data).

Problem definition. Given two $node \times feature$ matrices, $\mathcal{F}^{anon}$ corresponding to the anonymized graph $\mathcal{G}^{anon}$ and $\mathcal{F}^{aux}$, corresponding to the auxiliary graph $\mathcal{G}^{aux}$, map each node in $\mathcal{G}^{anon}$ to its I^+ set containing nodes in $\mathcal{G}^{aux}$.

We assume that graphs $\mathcal{G}^{anon}$ and $\mathcal{G}^{aux}$ contain a non-empty set of common nodes. $\mathcal{G}^{anon}$ may be a graph that is released by an organization after applying an anonymization technique (e.g., edge perturbation). $\mathcal{G}^{aux}$ may be a graph collected at different points in time or by different sources. In this paper, we use the terms I^+ sets and clusters interchangeably.

Problem requirements. There are two requirements. The first is scalability. We do not want to compare each row in $\mathcal{F}^{anon}$ with every row in $\mathcal{F}^{aux}$ (i.e., we would like to avoid the dreaded quadratic comparison). The second is a nonparametric solution. We do not want to assume *a priori* what the size of

This work was supported in part by NSF CNS-1314603, by DTRA HDTRA1-10-1-0120, and by DAPRA under SMISC Program Agreement No. W911NF-12-C-0028.

[1] Similar to other studies, the adversary (who wants to comprise the identity of individuals) can access auxiliary graphs, from which he/she can compute the same structural features as the ones released with the anonymized data [3].

Graph Type	Noise Type	Avg. *Recall*	Avg. *Precision* of Recalled Nodes
Synthetic	Known	76%	64%
Real	Known	69%	80%
Real	Unknown	37%	74%

TABLE I. HIGHLIGHTS OF OUR FINDINGS. *Recall* IS THE FRACTION OF NODES FOR WHICH THE CORRECT I^+ HAS BEEN IDENTIFIED. *Precision* IS THE SIZE OF THE SET THAT HAS BEEN ELIMINATED AS A LIKELY MATCH. *Precision* AND *Recall* ARE IN $[0\%, 100\%]$ WITH HIGHER VALUES BEING BETTER. FOR INSTANCE, IN REAL GRAPHS WITH UNKNOWN NOISE, 37% OF THE POPULATION ON AVERAGE CAN BE IDENTIFIED AND ON AVERAGE THEIR IDENTITY CAN BE NARROWED DOWN TO 26% OF THE POPULATION.

I^+ is; instead the solution to the problem must automatically find the correct size for I^+ for each anonymized individual.

To solve the aforementioned problem, we introduce a new objective function and a novel method, called $RRID^+$ (short for *Recursive Re-Identification*$^+$). Our method, $RRID^+$, is a greedy approach that utilizes hierarchical paired clustering by median. $RRID^+$ recursively clusters and matches subsets of nodes from $\mathcal{F}^{anon}$ to $\mathcal{F}^{aux}$, until the precision of the matches starts to degrade. Our experiments compare the performance of $RRID^+$ to other approaches on various real and synthetic graphs with known (synthetic) and unknown (real) noise. Moreover, we conduct a feature-selection study of $RRID^+$ and analyze its performance in terms of runtime and in terms of the distribution of its $I+$ sets.

The **contributions** of our work are as follows: **(1)** We define the problem of threatening an individual's privacy as a relative concept. **(2)** We study the problem of threatening an individual's privacy based on *only* a handful of local structural features. **(3)** We propose $RRID^+$: a novel, scalable, and nonparametric approach, which narrows down the identity of each anonymized individual to his/her I^+ set. These $I+$ sets are clusters that $RRID^+$ identifies collectively. **(4)** We present an extensive empirical study that includes a variety of real and synthetic graph, numerous noise models, feature selection, and analysis of various noise models.

The outline of the paper is as follows. We present related works next. In Section III, we describe some preliminaries, our objective function, and $RRID^+$. In Section IV, we discuss our data sets, experimental setup and methodology, and results. We conclude the paper with a summary in Section V.

II. RELATED WORK

To the best of our knowledge, we are the first to study the privacy of anonymized nodes, when only their structural features are released. Below, we summarize related research.

Node Re-identification in Networks. This problem has been studied extensively. Backstrom et al. [4] show that by knowing the identities of a subset of nodes and the subgraph induced by them, a passive attacker can reveal identities of neighboring nodes. Narayanan and Shmatikov [1] demonstrate the feasibility of large-scale re-identification of social networks by using a set of seed nodes (i.e., a set of nodes whose identity is known) and taking advantage of the sparsity of the graphs in the anonymized network. Korula and Lattanzi [5] show that edges between graphs can be matched by assuming that there exists an underlying graph from which both graphs have been generated. Zhu et al. [6] first find mappings between highly probable node-pairs and then use them as seeds to find the identities of the remaining nodes. Hay et al. [7] illustrate that an adversary could re-identify nodes through structural queries and propose an algorithm that anonymizes a graph by partitioning the nodes into groups. Henderson et al. [8] present an algorithm, called $ReFeX$, which uses recursively extracted structural features to re-identify nodes across graphs. Pedarsani and Grossglauser [9] describe conditions under which a random graph could be deanonymized. Their finding implies that: given two graphs, it is feasible to re-identify nodes using the structural similarity of nodes. We assume no knowledge of the corresponding identities across the graphs and rely solely on local structural features of nodes.

Nearest Neighbor (NN) Search. Commonly used NN approaches such as Locality Sensitive Hashing (LSH) [10] and KD-trees [11] are not suitable for our problem. Such methods rely on the number of nearest neighbors, k, to be given to them *a priori*. Our problem requires k to be selected automatically for each anonymized node. Nonetheless, in Section IV we compare our method to LSH and KD-trees by providing the values of k that our method $RRID^+$ finds.

Graph Matching. Our work is different from graph matching because we assume that the anonymized graph's connectivity (i.e., its adjacency matrix) is not released. Reconstructing a graph's connectivity from a its structural features is a hard combinatorial task.

Adding Noise to Graphs. The most common methods for adding noise to a given graph $\mathcal{G}$ are (a) removing nodes, (b) removing edges, or (c) rewriting edges while maintaining the same degree distribution. Some studies [5], [9] assume the presence of an underlying graph $\mathcal{G}$ and introduce noise by adding or removing edges from $\mathcal{G}$ to get $\mathcal{G}_1$ and $\mathcal{G}_2$. While others [12], [13] add noise to the adjacency matrix over a random permutation of the nodes. In Section IV, we use the aforementioned methods to add synthetic (known) noise to $\mathcal{G}$ and generate $\mathcal{G}^{aux}$.

Anonymization in Networks. We view research on anonymization in networks as related but tangential to our work. Our work is about *deanonymization* in networks, where for each anonymized node we return a small set of known nodes that is most likely to include the anonymized node. We use random edge perturbation [14] to anonymize the original graph (see Section IV-B).

III. PROPOSED METHOD

This section is divided into three parts: (1) preliminaries, (2) objective function, and (3) the $RRID^+$ algorithm. Table II lists the notation used in the paper.

A. Preliminaries

We assume that the link structure of the anonymized graph $\mathcal{G}^{anon}$ is not released to the public. Instead, a handful of structural features are released for each node in $\mathcal{G}^{anon}$. These structural features capture local information for each node. They are: (1) node degree, (2) node clustering coefficient, (3) average degree of a node's neighbors, (4) average clustering coefficient of a node's neighbors, (5) number of edges in a

v	a node in a graph
$\mathcal{G}^{anon}$	anonymized graph
$\mathcal{F}^{anon}$	structural feature table of $\mathcal{G}^{anon}$
n	# of rows in $\mathcal{F}^{anon}$ (i.e., # of nodes in $\mathcal{G}^{anon}$)
f	# of columns in $\mathcal{F}^{anon}$ (i.e., # of structural features)
$\mathcal{G}^{aux}$	auxiliary graph
n^{aux}	# of nodes in the auxiliary graph
m^{aux}	# of edges in the auxiliary graph
$\mathcal{F}^{aux}$	feature table for $\mathcal{G}^{aux}$ with n^{aux} rows and f columns
C_j^i	the j^{th} cluster of $\mathcal{F}^{anon}$ at recursion-level i
$\lvert C_j^i \rvert$	# of nodes in cluster C_j^i
$C_j^{i,aux}$	the j^{th} cluster of $\mathcal{F}^{aux}$ at recursion-level i
$\lvert C_j^{i,aux} \rvert$	# of nodes in cluster $C_j^{i,aux}$

TABLE II. NOTATIONS USED IN THE PAPER

node's ego network,[2] (6) number of outgoing edges from the node's ego network, and (7) number of neighbors of a node's ego network. These seven features correspond to four social theories−namely, Social Capital, Structural Hole, Balance, and Social Exchange [15]. Thus, for the anonymized graph, we have a table $\mathcal{F}^{anon}$ whose rows correspond to nodes in $\mathcal{G}^{anon}$ and whose columns corresponds to these structural features.

The adversary extracts the aforementioned seven features for the nodes in a publicly available auxiliary graph $\mathcal{G}^{aux}$ and produces a feature table $\mathcal{F}^{aux}$. The goal of the adversary is as follows. For each node v in $\mathcal{G}^{anon}$, find a small set of the most likely nodes in $\mathcal{G}^{aux}$ that match v. The adversary's underlying assumption is that if a node $v \in \mathcal{G}^{anon}$ has a corresponding node $v' \in \mathcal{G}^{aux}$, then v and v' are structurally similar. That is, their rows in $\mathcal{F}^{anon}$ and $\mathcal{F}^{aux}$ are similar. Wanting a robust approach, the adversary does not assume an *a priori* threshold on the similarity between matches in $\mathcal{F}^{anon}$ and $\mathcal{F}^{aux}$. In addition, he/she does not assume a pre-specified number k for the top-k most likely matches of any node v in $\mathcal{G}^{anon}$.

B. Objective Function

The adversary's objective function is similar to objective functions from Information Retrieval−i.e., match each node in $\mathcal{G}^{anon}$ to a small set of most likely nodes in $\mathcal{G}^{aux}$. In other words, for each node v in $\mathcal{G}^{anon}$, the adversary wants high *Recall* and high *Precision* w.r.t. the auxiliary graph $\mathcal{G}^{aux}$. An efficient and effective way of achieving this objective is to recursively cluster $\mathcal{F}^{anon}$ and $\mathcal{F}^{aux}$ and match their clusters (details in the next section). The challenge resides in finding matching clusters in $\mathcal{F}^{anon}$ and $\mathcal{F}^{aux}$ where both *Recall* and *Precision* are high.

1) ***Recall and Precision****:* In our setting, *Recall* for a node v in $\mathcal{F}^{anon}$ is 1 if the selected cluster $C_j^{i,aux}$ in $\mathcal{F}^{aux}$ contains v. Otherwise, it is 0. Formally, we have:

$$\forall v \in \mathcal{F}^{anon} : Recall(v, \mathcal{F}^{aux}) = \begin{cases} 1, & \text{if } v \in C_j^{i,aux} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Here $C_j^{i,aux}$ is the j^{th} cluster of $\mathcal{F}^{aux}$ at recursion-level i.

Then, *Recall* for the anonymized graph $\mathcal{G}^{anon}$ (represented by $\mathcal{F}^{anon}$) given auxiliary graph $\mathcal{G}^{aux}$ (represented by $\mathcal{F}^{aux}$) is the sum of the individual node *Recall* values divided by the number of nodes in the anonymized graph−i.e., the average *Recall* of the nodes in the anonymized graph.

$$Recall(\mathcal{F}^{anon}, \mathcal{F}^{aux}) = \frac{\sum_{v \in \mathcal{F}^{anon}} Recall(v, \mathcal{F}^{aux})}{n} \in [0, 1] \quad (2)$$

Precision for a node v in $\mathcal{F}^{anon}$ is defined as the fraction of nodes that can be disregarded from being likely matches for v in $\mathcal{F}^{aux}$. Formally, we have:

$$\forall v \in \mathcal{F}^{anon} : Precision(v, \mathcal{F}^{aux}) = \begin{cases} 1 - \frac{|C_j^{i,aux}|}{n^{aux}}, \in [0, 1) & \text{if } v \in C_j^{i,aux} \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

Some properties of *Precision* are as follows:

- $Precision(v, \mathcal{F}^{aux}) = 0$ when node v either is incorrectly matched or does not exist in $\mathcal{F}^{aux}$.
- $Precision(v, \mathcal{F}^{aux}) = 1 - \frac{1}{n^{aux}}$ when node v is matched exactly to its corresponding node in $\mathcal{F}^{aux}$ (i.e., $C_j^{i,aux}$ only contains v).
- $Precision(v, \mathcal{F}^{aux}) < 1$ because $|C_j^{i,aux}| > 0$.

Similar to *Recall*, *Precision* for the anonymized graph $\mathcal{G}^{anon}$ (represented by $\mathcal{F}^{anon}$) given the auxiliary graph $\mathcal{G}^{aux}$ (represented by $\mathcal{F}^{aux}$) is the sum of the individual node *Precision* values divided by the number of nodes in the anonymized graph−i.e., the average *Precision* of the nodes in the anonymized graph. Hence, we have:

$$Precision(\mathcal{F}^{anon}, \mathcal{F}^{aux}) = \frac{\sum_{v \in \mathcal{F}^{anon}} Precision(v, \mathcal{F}^{aux})}{n} \in [0, 1) \quad (4)$$

2) ***Precision as a Function of Recall****:* Looking at Equations 1 and 3, we can rewrite *Precision* for a node v in $\mathcal{F}^{anon}$ as follows:

$$\forall v \in \mathcal{F}^{anon} : Precision(v, \mathcal{F}^{aux}) = Recall(v, \mathcal{F}^{aux}) \times \left(1 - \frac{|C_j^{i,aux}|}{n^{aux}}\right) \quad (5)$$

Moreover, since the clusters in $\mathcal{F}^{anon}$ are disjoint, we can rewrite Equation 4 as the sum of the *Precision* values of nodes in each cluster of $\mathcal{F}^{anon}$ (see Equation 6).

$$Precision(\mathcal{F}^{anon}, \mathcal{F}^{aux}) = \frac{1}{n} \sum_{j=1}^{\#\text{ clusters in } \mathcal{F}^{anon}} \sum_{v \in C_j^i} (Recall(v, C_j^{i,aux}) \times (1 - \frac{|C_j^{i,aux}|}{n^{aux}})) \quad (6)$$

Thus, the adversary's objective is to find *matched* clusterings of $\mathcal{F}^{anon}$ and $\mathcal{F}^{aux}$ that maximize $Precision(\mathcal{F}^{anon}, \mathcal{F}^{aux})$ in Equation 6.

3) ***Precision and Recall Between Clusters****:* If the adversary is going to recursively cluster $\mathcal{F}^{anon}$ and $\mathcal{F}^{aux}$ and match the resulting clusters, he/she needs to compute *Recall* and *Precision* between two clusters. The unnormalized *Recall*

[2] A node's ego network is its 1-hop induced subgraph.

between two clusters C_j^i and $C_j^{i,aux}$ measures their node overlap. Formally, we have:

$$Recall(C_j^i, C_j^{i,aux}) = \sum_{v \in C_j^i} Recall(v, C_j^{i,aux}) \quad (7)$$

$$= |C_j^i \cap C_j^{i,aux}| \in \left[0, |C_j^i|\right] \quad (8)$$

The unnormalized *Precision* between two clusters C_j^i and $C_j^{i,aux}$ is defined as the sum of the individual node *Precision* values. Formally, we have:

$$Precision(C_j^i, C_j^{i,aux}) = \sum_{v \in C_j^i} Precision(v, C_j^{i,aux})$$

$$= \sum_{v \in C_j^i} Recall(v, C_j^{i,aux})(1 - \frac{|C_j^{i,aux}|}{n^{aux}})$$

$$\in \left[0, |C_j^i \cap C_j^{i,aux}|\right) \quad (9)$$

*4) **Estimating Precision and Recall**:* Since the adversary does not have the node correspondences between the two graphs, he/she has to estimate *Recall*, which in turn provides an estimate for *Precision*.[3] In particular, he/she needs to estimate *Recall* between two clusters. We define the estimated *Recall* between two clusters to be a function of the estimated *Recall* of the parent clusters, the size of the cluster C_j^i, and the relative distance between the centroids of C_j^i and $C_j^{i,aux}$. The relative distance between the centroids of C_j^i and $C_j^{i,aux}$ is a proxy for the similarity between the two clusters. Furthermore, this is weighted by the estimated recall of the parent cluster C^{i-1} and the size of C_j^i. The estimated *Recall* value is in the interval $[0, 1]$. It is at its maximum value when the distance between the cluster centroids is 0 and the estimated *Recall* of its parent clusters is 1. Formally, we have:

$$\hat{R}ecall(C_j^i, C_j^{i,aux}) = \hat{R}ecall(C^{i-1}, C^{i-1,aux}) \times |C_j^i| \times (1 - NormalizedDist(C_j^i, C_j^{i,aux})) \quad (10)$$

The $NormalizedDist$ between two clusters is the normalized distance between their centroids. Formally, we have:

$$NormalizedDist(C_j^i, C_j^{i,aux}) = \frac{Dist(Centroid(C_j^i), Centroid(C_j^{i,aux}))}{\sum_k Dist(Centroid(C_k^i), Centroid(C_k^{i,aux}))} \quad (11)$$

We use *Canberra* distance[4] to measure the distance between cluster centroids, since Canberra distance is sensitive around zero and normalizes the absolute difference of the individual comparisons.

C. $RRID^+$ Algorithm

Figure 1 provides an overview of our algorithm. $RRID^+$ takes $\mathcal{F}^{anon}$ and $\mathcal{F}^{aux}$ and employs a recursive, greedy, top-down cluster-and-match approach. $RRID^+$ terminates when it can no longer improve its objective function– i.e., the estimated *Precision* in Equation 6.

[3]Note that *Precision* is defined in terms of *Recall*. See Equation 5.

[4]$Canberra(\vec{p}, \vec{q}) = \sum_{i=1}^{d} \frac{|p_i - q_i|}{|p_i| + |q_i|}$, where $d = |\vec{p}| = |\vec{q}|$.

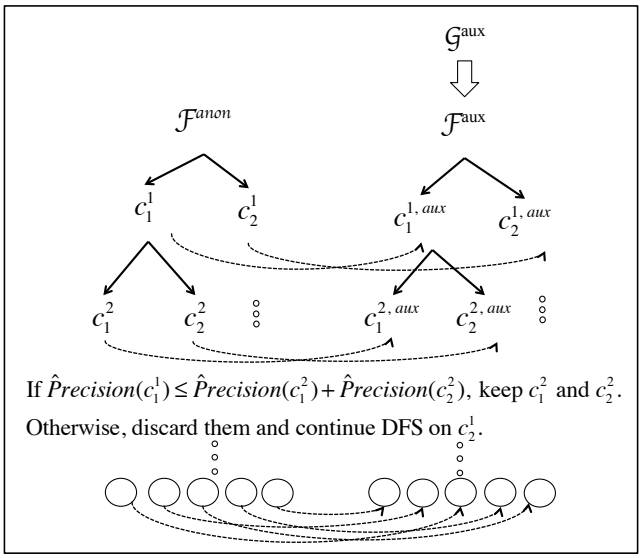


Fig. 1. Overview of our $RRID^+$ Algorithm. At each level of the recursion, clusters are matched across the two structural feature tables. Recursion stops when the estimated *Precision* (see Equation 6) does not improve.

*1) **Clustering**:* Recall that the $\mathcal{F}^{anon}$ and $\mathcal{F}^{aux}$ feature tables each have seven features (described in Section III-A). We use the median of the feature with the highest standard deviation in $\mathcal{F}^{anon}$ to cluster the graphs into two clusters. Thus, as shown in Figure 1, we get C_1^1 and C_2^1 from $\mathcal{F}^{anon}$ and $C_1^{1,aux}$ and $C_2^{1,aux}$ from $\mathcal{F}^{aux}$. At every subsequent iteration, the clusters are again split based on the feature with the highest standard deviation in clusters of $\mathcal{F}^{anon}$. We continue splitting the data until the stopping condition (described below) is met. Note that the binary trees generated by the clustering of $\mathcal{F}^{anon}$ and $\mathcal{F}^{aux}$ is similar to the KD-tree, except that we match the clusters across the binary trees at every level. Note that, for the sake of simplicity, we chose to split the features tables into two clusters. We could easily extend the method to consider more than two clusters at each level of recursion.

*2) **Matching**:* At every level of the recursion, we match a cluster from $\mathcal{F}^{anon}$ to a cluster from $\mathcal{F}^{aux}$ such that the total distance between the centroids of the matched clusters is minimized. For example in Figure 1, a matching of C_1^1 to $C_1^{1,aux}$ and of C_2^1 to $C_2^{1,aux}$ would occur only if the following condition was satisfied:

$$Dist(Centroid(C_1^1), Centroid(C_1^{1,aux})) + Dist(Centroid(C_2^1), Centroid(C_2^{1,aux})) \leq Dist(Centroid(C_1^1), Centroid(C_2^{1,aux})) + Dist(Centroid(C_2^1), Centroid(C_1^{1,aux})) \quad (12)$$

As described in the previous section, we use the Canberra distance for our distance function.

*3) **Terminating the Recursion**:* As shown in Figure 1, after clustering and matching a pair of clusters, we need to determine if splitting the clusters further increases the estimated *Precision*. Specifically, we check whether $\hat{P}recision(C_j^i, C_j^{i,aux}) \leq \sum_k \hat{P}recision(C_k^{(i+1)}, C_k^{(i+1),aux})$. For the clusters that meet the aforementioned condition (i.e., they have a higher total estimated *Precision* value at level $i + 1$), the recursion

continues. For the clusters that have a lower total estimated *Precision*, we terminate the recursion at level i and continue the recursion at a different brunch.

4) ***Computational Complexity****:* Our approach clusters the feature tables recursively by the median of the features, thus splitting the data into two equal-sized clusters at every step. It takes $O(f \times n \times \log\ n)$ time to build such a binary tree, using heapsort to find the median, where f is the number of features. At each split (i.e., each time we find a median), we check for the stopping condition. At each level of the tree, it takes $O(n)$ to compute the stopping condition, which includes computing the centroid and distances for the estimated *Precision*. The depth of the tree is at most $\log\ n$. So, the total time taken by $RRID^{+}$ is $O(f \times n \times \log\ n) + \log\ n \times O(n) \approx O(n \times \log\ n)$ since $f << n$.

IV. Experiments

A. Data Description

We ran experiments on synthetic and real graphs with known (synthetic) and unknown (real) noise. All of our graphs are undirected and unweighted.

We generated **synthetic graphs** by using the following four graph models. **(S1)** *Barabási-Albert Preferential Attachment* graph model [16]: A new node preferentially attaches itself to 5% of the nodes. **(S2)** *Erdös-Rényi Random* graph model [17]: We use the $G(n, p)$ generator, where p is 0.01. **(S3)** *Forest-Fire* graph model [18]: We set forward-burning probability to 0.5 and backward-burning probability to 0.56. **(S4)** *Watts-Strogatz Small-World* graph model [19]: We set degree to 50 and rewiring probability to 0.1.

For each graph model, we generated 10 graphs with 5K nodes using the aforementioned model parameters. This process yielded synthetic graphs where the average degree for each graph was about 50. We used iGraph[5] to generate the Forest-Fire graphs and NetworkX[6] to generate the rest.

Our **real graphs** include four different sets of graphs. Table III lists their basic statistics. We have two communication graphs and two social graphs. **(R1)** *Twitter Retweets*: collected from May to September 2009. A node is a Twitter user and an edge between two nodes indicates that either of the users retweeted the other user's tweet during the period of data collection. We break the data into 5 graphs over time, one for each month from May to September 2009. **(R2)** *Yahoo! IM*:[7] collected over 28 days in April 2008. We divide the data into four graphs: each covering a week in April 2008. **(R3)** *DBLP Computer Science Bibliography*:[8] collected from 2005 to 2009. We extract the co-authorship graph from KDD, SDM, ICDM, CIKM, SIGMOD, and VLDB. We break the data into five graphs over time: one graph for each year from 2005 to 2009. **(R4)** *IMDB Movie Collaborations*:[9] collected from 1950 to 1955. We extracted graphs of all individuals credited in movies made during 1950 to 1955. An edge between two individuals exist if they have been credited for a common movie. We divide the data into six graphs: one for each year.

[5]http://igraph.sourceforge.net/

[6]http://networkx.github.io/

[7]http://sandbox.yahoo.com/

[8]http://www.informatik.uni-trier.de/~ley/db/

[9]http://www.imdb.com/

B. Experimental Setup and Methodology

Figure 2 shows the three different settings of graph- and noise-types used in our experiments. Section IV-B1 describes the structural features calculated. *Edge perturbation*, shown in Figure 2, is a form of graph anonymization that is described in Section IV-B2. In Section IV-B3, we report the noise added to $\mathcal{G}^{aux}$, as shown in Figures 2(b) and (c). Section IV-B4 refers to Figure 2(a). Section IV-B5 describes the competing methods.

1) ***Structural Features****:* Recall that the adjacency matrix of the anonymized graph is not released; instead the anonymized graph is given to us in terms of seven node- and ego-net level structural features, which were described in Section III-A. We extract the same seven features for the auxiliary graph.

2) ***Graph Anonymization****:* We assume that the released $\mathcal{F}^{anon}$ is the structural feature table of the anonymized graph $\mathcal{G}^{anon}$. It has been shown that anonymization by removing the unique identifiers alone can risk the privacy of the dataset [1], [20]. Here, we anonymize the graphs by applying random edge perturbation described in [14]. Random edge perturbation randomly removes a fixed fraction of the edges, then adds the same number of edges back into the graph at random. We ran experiments with varying levels of perturbation from 0.01 to 1; and found that at perturbation values higher than 0.1, the graphs gradually begin to lose their structural characteristics. We omit these results for the sake of brevity. In the experiments reported here, anonymized graphs were obtained by randomly perturbing 10% of the edges.

3) ***Known Noise Models****:* We discussed various noise models used in the literature in Section II. Given a graph $\mathcal{G}^{aux}$, (either synthetic or real), we add three different known noise models to it in order to generate variants of it ($\mathcal{G}^{anon}$). We use the following three known noise models: **(1)** *random edge deletion*, where edges are randomly deleted with probability p; **(2)** *random node deletion*, where nodes are randomly deleted with probability p (when a node is deleted, all of its edges are also deleted); and **(3)** *random rewiring*, where edges are randomly rewired with probability p; but the graph retains its original degree distribution. In our experiments, we use three different values for noise parameter p: 0.05, 0.1, and 0.2.

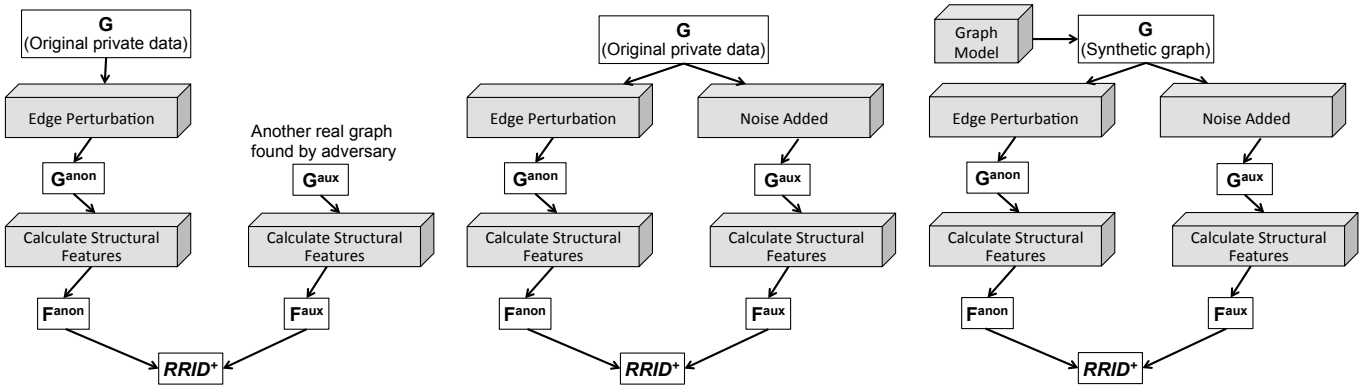
4) ***Unknown Noise Models****:* In real-world settings, we do *not* know the noise model between the anonymized graph $\mathcal{G}^{anon}$ and the auxiliary graph $\mathcal{G}^{aux}$. We run experiments where we select one of our real graphs as $\mathcal{G}^{aux}$ and another one as $\mathcal{G}^{anon}$ using the following settings:

- Retweet graph of May 2009 is used as $\mathcal{G}^{aux}$. The edge-perturbed retweet graphs from June to September 2009 are, in turn, used as $\mathcal{G}^{anon}$.
- Yahoo! IM graph from Week 1 is used as $\mathcal{G}^{aux}$. The edge-perturbed Yahoo! IM graphs from Weeks 2 to 4 are, in turn, used as $\mathcal{G}^{anon}$.
- DBLP 2005 graph is used as $\mathcal{G}^{aux}$. The edge-perturbed DBLP 2006 to 2009 are, in turn, used as $\mathcal{G}^{anon}$.
- IMDB 1950 graph is used as $\mathcal{G}^{aux}$. The edge-perturbed IMDB 1951 to 1955 are, in turn, used as $\mathcal{G}^{anon}$.

Table III shows the maximum *Recall* in these experiments, where maximum *Recall* is computed as the ratio of the number of overlapping nodes (in $\mathcal{G}^{anon}$ and $\mathcal{G}^{aux}$) to the number of nodes in $\mathcal{G}^{anon}$.

Real Graphs	Average # of Nodes (Std. Dev.)	Average # of Edges (Std. Dev.)	Average # of Connected Comps (Std. Dev.)	Average Size of LCC (Std. Dev.)	Average of Average Degrees (Std. Dev.)	Average Overlap $\mathcal{G}^{anon}$ & $\mathcal{G}^{aux}$ (Std. Dev.)
Twitter Retweet May to Sep. 2009 (Monthly)	64,072.6 (36,793.0)	81,906.8 (54,629.6)	4,815.4 (1,435.2)	0.7 (0.1)	2.5 (0.2)	0.15 (0.06)
Yahoo! IM April 2008 (Weekly)	84,992.3 (10,199.0)	261,167.8 (52,899.2)	851.0 (278.9)	0.8 (0.0)	6.1 (0.5)	0.86 (0.04)
DBLP Co-authorship 2005 to 2009 (Yearly)	2,045.2 (454.7)	4,024.8 (1,162.8)	341.8 (53.1)	0.2 (0.1)	3.9 (0.3)	0.24 (0.06)
IMDB Collaboration 1950 to 1955 (Yearly)	10,887.7 (473.8)	236,132.2 (54,173.5)	101.7 (19.3)	0.9 (0.0)	43.3 (8.8)	0.38 (0.04)

TABLE III. REAL GRAPHS USED IN OUR EXPERIMENTS. WE PAIR REAL GRAPHS OVER TIME TO TEST HOW WELL $RRID^+$ PERFORMS ON REAL GRAPHS WITH UNKNOWN NOISE MODELS. LCC IS SHORT FOR LARGEST CONNECTED COMPONENT. AVERAGE OVERLAP BETWEEN $\mathcal{G}^{anon}$ AND $\mathcal{G}^{aux}$ IS THE MAXIMUM ACHIEVABLE *Recall*.



(a) Real graphs with unknown noise

(b) Real graphs with known noise added to $\mathcal{G}^{aux}$

(c) Synthetic graphs with known noise added to $\mathcal{G}^{aux}$

Fig. 2. Graph- and noise-types used in our experiments. The real graphs used in (a) and (b) are listed in Table III and the Synthetic graphs used in (c) are described in Sec. IV-A.

*5) **Comparison with Competing Methods**:* We compare our method $RRID^+$ (described in Section III) to the following five methods: **(M1)** *k-means*, where we set k to be the number of clusters found by $RRID^+$; **(M2)** *random clustering*, where we randomly assign nodes to k clusters; again we set k to be the number of clusters found by $RRID^+$; **(M3)** *paired hierarchical random clustering*, where the procedure is the same as $RRID^+$ except that instead of clustering using median, we randomly assign nodes to clusters with the same cluster sizes as $RRID^+$; **(M4)** *Locality Sensitive Hashing* (LSH) [10]; and **(M5)** *KD-tree* [11]. Note that, all of the above methods require the input k to be specified. In order to perform a fair comparison, we set the input k as that found by $RRID^+$. The last two competing methods, LSH and KD-tree, are popular nearest-neighbor methods. Note that for the nearest neighbor methods the query set is as big as the search space. For the above methods, we evaluate the performance in terms of *Average Recall* and the *Average Precision* of the nodes that were correctly recalled. *Precision* of correctly recalled nodes elucidates the extent of re-identification and makes the results easier to understand. Note that when *Recall* is 0, then *Precision* is also 0 (see Equation 5).

$$\textit{Precision} \text{ of recalled nodes}(\mathcal{G}, \mathcal{G}_{aux}) = \frac{\sum_{v \in RecalledNodes} Precision(v, \mathcal{G}_{aux})}{|RecalledNodes|} \quad (13)$$

where $RecalledNodes = \{\forall v : Recall(v, G) = 1\}$. Note that *Precision* can be computed from *Precision* of recalled nodes and *Recall* as follows:

$$\textit{Precision}(\mathcal{G}, \mathcal{G}_{aux}) = \textit{Precision} \text{ of recalled nodes}(\mathcal{G}, \mathcal{G}_{aux}) \times \textit{Recall}(\mathcal{G}, \mathcal{G}_{aux}) \quad (14)$$

C. Results

We report and discuss the answers to several questions. *Recall* reported here is scaled based on the overlap between the graphs (see last column of Table III) and *Precision* is reported for the recalled nodes (see Equation 13). Thus, *Recall* and *Precision* of recalled nodes reported here are in $[0, 1]$. For brevity, we restrict the discussions of Q3 and Q4 to real graphs with known and unknown noise models.

Q1. How does our approach $RRID^+$ perform (in terms of *Precision* and *Recall*) when compared to other approaches? Table IV reports our comparative results in terms of average *Precision* of recalled nodes and average *Recall*. Across real graphs with unknown noise (i.e., in real-world settings), our approach $RRID^+$ performs as well as the nearest neighbor approaches. But, KD-tree and LSH have several drawbacks. First, they require k to be known *a priori*. Second, in the nearest neighbor approach, each node in $\mathcal{F}^{anon}$ is considered an independent search query; thus, the number of queries is as large as $\mathcal{G}^{anon}$. Our approach $RRID^+$ automatically selects the value for k and *collectively* matches all nodes in $\mathcal{F}^{anon}$ to

	Real Graph with Unknown (i.e., Real) Noise	Real Graph with Known (i.e., Synthetic) Noise	Synthetic Graph with Known (i.e., Synthetic) Noise
Jaccard Similarity	0.02	0.68	0.62
	Avg. Recall / Avg. Precision of Recalled nodes	Avg. Recall / Avg. Precision of Recalled nodes	Avg. Recall / Avg. Precision of Recalled nodes
$RRID^+$ **(our method)**	**0.37 / 0.74**	**0.69 / 0.80**	**0.76 / 0.64**
K-means clustering	0.09 / 0.79	0.14 / 0.76	0.31 / 0.64
Random clustering	0.11 / 0.68	0.11 / 0.76	0.21 / 0.64
Paired hierarchical random clustering	0.16 / 0.78	0.14 / 0.81	0.31 / 0.64
KD-Tree	0.39 / 0.74	0.77 / 0.80	0.69 / 0.63
LSH	0.37 / 0.75	0.76 / 0.81	0.66 / 0.64

TABLE IV. AVERAGE *Recall* AND *Precision* OF RECALLED NODES OF OUR METHOD $RRID^+$ (IN BOLDFACE) AS COMPARED TO FIVE COMPETING AND BASELINE METHODS. THE *Recall* REPORTED HERE ARE SCALED BASED ON THE OVERLAP OF NODES BETWEEN $\mathcal{G}^{anon}$ AND $\mathcal{G}^{aux}$. IN REAL GRAPHS WITH UNKNOWN NOISE, THE MAXIMUM *Recall* IS SHOWN IN TABLE III. THE *Precision* REPORTED HERE IS THAT OF THE RECALLED NODES, AS GIVEN IN EQ.13. WE ALSO REPORT THE *Jaccard Similarity* BETWEEN $\mathcal{G}^{anon}$ AND $\mathcal{G}^{aux}$, WHICH INDICATES THE HARDNESS OF THE TASK AT HAND. THE LOWER THE *Jaccard Similarity*, THE HARDER THE TASK. $RRID^+$ BEATS THE BASELINE METHODS AND IS COMPARABLE TO THE COMPETING METHODS KD-TREE AND LSH. BUT, $RRID^+$ HAS TWO ADVANTAGES OVER KD-TREE AND LSH: (1) IT IS NONPARAMETRIC AND (2) IT HAS FASTER RUNTIMES (SEE TABLE V).

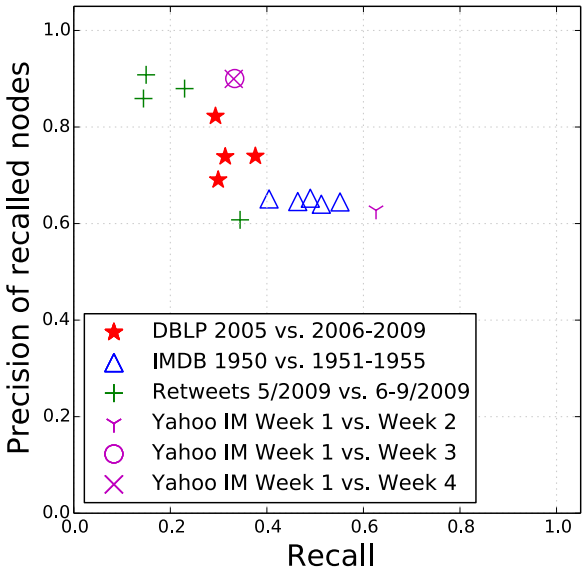


Fig. 3. (Best viewed in color) $RRID^+$ on real graphs with unknown noise: *Precision* of recalled nodes vs. *Recall*. The higher the *Recall* and *Precision* of recalled nodes, the more privacy is threatened. Here we do not know the noise model. In this setting, $RRID^+$ detects small-sized clusters (i.e., the median cluster size is a small fraction of all the nodes in $\mathcal{G}^{aux}$). Such small-sized clusters have lower Recall because they are less likely to contain a particular individual. But when the correct cluster is selected, the *Precision* of recalled nodes is high because the cluster size is small.

nodes in $\mathcal{F}^{aux}$. The known (i.e., synthetic) noise models here are deleting nodes, deleting edges, and rewiring edges while maintaining the same degree distribution.

Q2. What is the relationship between *Recall* and *Precision* of recalled nodes (a) when the graphs are real and the noise model between $\mathcal{G}^{anon}$ and $\mathcal{G}^{aux}$ is unknown, (b) when the graphs are real and the noise model is known, and (c) when the graphs are synthetic and the noise model is known? As expected, in all settings, lower *Recall* often leads to higher *Precision* of recalled nodes. In Figure 3, we do not know the noise model; both $\mathcal{G}^{anon}$ and $\mathcal{G}^{aux}$ are real graphs (e.g., $\mathcal{G}^{aux}$=DBLP 2005 and $\mathcal{G}^{anon}$=DBLP 2006; see Section IV-B4 for the pairings). In this setting, $RRID^+$ detects small-sized clusters (in order to capture the complexities of real-world noise). That is, the median cluster size is a small fraction of the nodes in graph $\mathcal{G}^{aux}$. When we have lower *Recall*, we have small-sized clusters that are less likely to contain a particular individual. However, when such clusters are also the correct clusters, then the *Precision* of recalled nodes is high because the cluster sizes are small.

Q3. How varied are the distributions of cluster sizes identified by $RRID^+$? How do these numbers compare to a baseline approach? For real graphs with unknown noise, the average number of clusters returned is 8.12 (std. dev is 4.47, median is 6.5). On average, the cluster size is 0.12 fraction of nodes in the $\mathcal{G}^{aux}$ (std. dev. is 0.12). We observe that $RRID^+$ detects clusters of varying sizes, with few outliers (i.e., very large sized clusters), and low values for median cluster sizes. This is an important observation since privacy in our setting is threatened by associating anonymized individuals to clusters of different sizes. For example, when $\mathcal{G}^{aux}$ is Yahoo! IM Week 1 and $\mathcal{G}^{anon}$ is Yahoo! IM Week 3, $RRID^+$ finds 15 clusters, the smallest cluster in $\mathcal{G}^{aux}$ has 1185 nodes (i.e., 0.01 fraction of the 82308 nodes in Yahoo! IM 1), the maximum cluster has 11,537 nodes (i.e., 0.14 fraction of the 82308 nodes in Yahoo! IM 1). The median cluster size is 5227 (i.e., 0.06 fraction of the 82308 nodes in Yahoo! IM 1). These results enable the adversary to partition anonymized individuals into various groups. Individuals in smaller sized clusters are more "distinguished" than individuals in larger sized clusters.

Q4. How do various measures of noise affect the performance of $RRID^+$? We quantify the noise between $\mathcal{G}^{anon}$ and $\mathcal{G}^{aux}$ in two ways. **(1)** *Lookalikes* between $\mathcal{F}^{anon}$ and $\mathcal{F}^{aux}$: For each node v in $\mathcal{F}^{anon}$, we define *Lookalikes* as the fraction of nodes in $\mathcal{F}^{aux}$ whose structural feature-vector distances (in terms of Canberra distance) is smaller than the structural feature vector distance of v to its equivalent node in $\mathcal{G}^{aux}$. *Lookalikes* between $\mathcal{F}^{anon}$ and $\mathcal{F}^{aux}$, is the average *Lookalikes* of nodes in $\mathcal{F}^{anon}$. **(2)** *Jaccard Similarity* between $\mathcal{G}^{anon}$ and $\mathcal{G}^{aux}$: This measure is defined as the normalized overlap between the edge-sets of every node across the two graphs $\mathcal{G}^{anon}$ and $\mathcal{G}^{aux}$. Note that *Jaccard Similarity* measures the ratio of common neighboring nodes (i.e., edge overlap)

	$RRID^+$ (our) (method)	KD-tree	LSH
Synthetic graphs with known (synthetic) noise			
Barabasi (5000 nodes)	**1.08 sec**	4.17 sec	1 min, 10 sec
Erdos-Renyi (5000 nodes)	**1.06 sec**	4.34 sec	1 min, 41 sec
Watts-Strogatz (5000 nodes)	**1.03 sec**	5.53 sec	1 min, 57 sec
Forest-fire (5000 nodes)	**1.05 sec**	4.76 sec	53.02 sec
Real graphs with known (synthetic) noise			
DBLP (2045.2 nodes)	**0.39 sec**	0.57 sec	11.20 sec
IMDB (10887.7 nodes)	**2.41 sec**	16.25 sec	2 min, 43 sec
Retweets (64072.6 nodes)	**6.71 sec**	54.56 sec	26 min, 27 sec
Yahoo IM (84992.3 nodes)	**40.53 sec**	15 min, 20 sec	6 hrs, 24 min
Real graphs with unknown (real) noise			
DBLP (2045.2 nodes)	**0.85 sec**	0.85 sec	14.75 sec
IMDB (10887.7 nodes)	**3.67 sec**	20.43 sec	3 min, 3 sec
Retweets (64072.6 nodes)	**26.65 sec**	3 min, 30 sec	1 hr, 25 min
Yahoo! IM (84992.3 nodes)	**38.59 sec**	22 min, 14 sec	3 hrs, 24 min

TABLE V. RUNTIME FOR $RRID^+$ AND THE NEAREST NEIGHBOR APPROACHES. THE NEAREST NEIGHBOR APPROACHES TAKE LONGER THAN $RRID^+$ SINCE IN THIS PROBLEM THE NUMBER OF QUERIES IS AS LARGE AS THE SEARCH SPACE, WHICH IS THE NUMBER OF NODES IN THE ANONYMIZED GRAPH (SHOWN IN PARENTHESES IN THE FIRST COLUMN).

between the corresponding nodes v and v^{aux} across the two graphs. Since $RRID^+$ operates on structural feature matrices, it performs well even when the corresponding nodes may have different neighbors but are structurally similar.

In our social graphs (DBLP and IMDB), *Precision* is negatively correlated with *Lookalikes*. We observe that higher *Lookalikes* tend to be associated with lower *Precision*. However, a lower *Lookalikes* value does not automatically imply a higher *Precision*. None of these measures of noise by themselves indicate how well an adversary can perform in terms of *Precision* and *Recall*. As a collection, however, these noise measures provide a loose guideline to when performance can be hindered: low *Jaccard Similarity* and high *Lookalikes* together can adversary affect *Precision* and *Recall*. A take-away message here is that there does not exist one equation that accurately quantifies noise for all real-world graphs because the underlying process generating real-world graphs are complex; and so are real-world noise models.

Q5. How does $RRID^+$ compare to other methods in terms of runtime? We implemented all algorithms in Matlab and ran them on CentOS machines with 3.0 x 8 GHz and 16 GB memory, running Linux 2.6. Table V shows runtime of $RRID^+$, LSH, and KD-tree measured in seconds. $RRID^+$ has lower runtime than that of KD-Tree and LSH for all datasets. The nearest neighbor methods have longer runtimes because their number of queries is large−namely, the number of queries equals the number of nodes in the anonymized graph. Across all methods, as the sizes of the graphs increase, the runtimes increase.

Q6. What is the effect of various subsets of structural features? In real graphs, regardless of the type of noise, using only degree-based features of a node's ego-net improves the *Precision* of recalled nodes compared to using all seven features described in Section III-A. This finding did not hold for synthetic graphs.

V. CONCLUSION

We presented $RRID^+$ (a scalable, nonparametric, hierarchical paired clustering by median approach) for distinguishing between individuals in an anonymized social graph, where only a handful of node-level structural features are released to the public. We demonstrated the performance of $RRID^+$ over a range of synthetic and real graphs and various known and unknown noise models. $RRID^+$ is scalable (linear on the number of edges) and nonparametric (where for each anonymized individual, $RRID^+$ returns a small and most likely set of known individuals for him/her). $RRID^+$ does not rely on edge-overlap or node-correspondences between the anonymized and auxiliary graphs.

REFERENCES

[1] A. Narayanan and V. Shmatikov, "De-anonymizing social networks," in *IEEE Symposium on Security and Privacy*, 2009, pp. 173–187.

[2] L. Sweeney, "Simple demographics often identify people uniquely," Data Privacy Working Paper 3, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep., 2000.

[3] P. Govindan, S. Soundarajan, and T. Eliassi-Rad, "Finding the most appropriate auxiliary data for social graph de-anonymization," in *1st KDD Workshop on Data Ethics*, 2014.

[4] L. Backstrom, C. Dwork, and J. Kleinberg, "Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural stenography," in *WWW*, 2007, pp. 181–190.

[5] N. Korula and S. Lattanzi, "An efficient reconciliation algorithm for social networks," *ArXiv e-prints*, Jul. 2013.

[6] Y. Zhu, L. Qin, J. X. Yu, Y. Ke, and X. Lin, "High efficiency and quality: large graphs matching," in *CIKM*, 2011, pp. 1755–1764.

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

[8] 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 *KDD*, 2011, pp. 663–671.

[9] P. Pedarsani and M. Grossglauser, "On the privacy of anonymized networks," in *KDD*, 2011, pp. 1235–1243.

[10] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," *Comm. of the ACM*, vol. 51, pp. 117–122, 2008.

[11] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Comm. of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[12] C. H. Q. Ding, T. Li, and M. I. Jordan, "Nonnegative matrix factorization for combinatorial optimization: Spectral clustering, graph matching, and clique finding," in *ICDM*, 2008, pp. 183–192.

[13] D. Koutra, H. Tong, and D. Lubensky, "Big-align: Fast bipartite graph alignment," in *ICDM*, 2013, pp. 389–398.

[14] F. Bonchi, A. Gionis, and T. Tassa, "Identity obfuscation in graphs through the information theoretic lens," *Inf. Sci.*, vol. 275, pp. 232–256, 2014.

[15] M. Berlingerio, D. Koutra, T. Eliassi-Rad, and C. Faloutsos, "Network similarity via multiple social theories," in *ASONAM*, 2013, pp. 1439–1440.

[16] A. Reka and Barabási, "Statistical mechanics of complex networks," *Reviews of Modern Physics*, vol. 74, pp. 47–97, 2002.

[17] P. Erdös and A. Rényi, "On random graphs I," *Publicationes Mathematicae Debrecen*, vol. 6, pp. 290–297, 1959.

[18] J. Leskovec, J. M. Kleinberg, and C. Faloutsos, "Graphs over time: densification laws, shrinking diameters and possible explanations," in *KDD*, 2005, pp. 177–187.

[19] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[20] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *IEEE Symposium on Security and Privacy*, 2008.