ISSN:0975-9646

Privacy Preservation of Social Network Data against Structural Attack using K-Auto restructure

V.Gnanasekar, S.Jayanthi

Department of Computer Science and Engineering
Anna University (BIT campus) - Tiruchirappalli.

Abstract— **Recent few years social networking sites are facing a problem of various attacks on their network which contains sensitive data. Usually social networks will publish their social network data for research purpose. Researchers and social network analysts can make use of these data to do research for decision making and market analysis. Before releasing the data for research, the social network site removes the identifiable parameters such as name, location, type of relationship, etc. Simply removing all identifiable personal information before releasing the data is insufficient. It is easy for an adversary to identify the target by performing different structural queries. Many of the previous studies were concentrated only on the anonymization part. We identify a special type of attack called structural attack. With the aim of resisting various structural attacks, in this paper, we proposed a new and efficient framework called k-Autorestructure which to protect against multiple structural attacks. There is no doubt in that our proposed algorithm will resist any kind of structural attack again the social network data.**

***Keywords*— Node Info, Link Info, Naively-Anonymized networks, Structural similarity, Auto restructure.**

I. Introduction

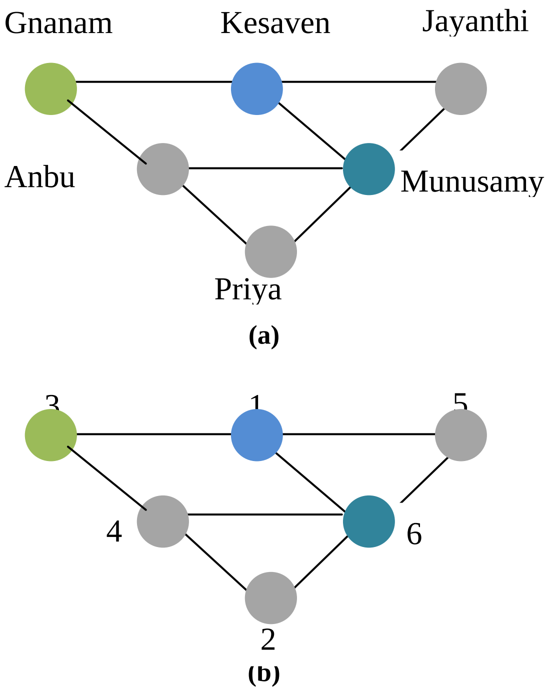
A network data set released from social network is a graph consists of a set of nodes and the edges between the nodes. Network data can be varied with different application areas. For e.g. a social network describes individuals and their personal relationships with other in the same network. Another example is an information network. An information network might describe a set of articles connected by citations. As the network graph is provided with different perspectives of information, networks can be analyzed in many ways: to study disease transmission, to measure the influence of a publication, and to evaluate the network's resiliency to faults and attacks. Such the studies reveal our understanding of network structure and function.

However, many networks contain highly sensitive data. For example, Facebook published a social network data which shows a set of individuals related by their relationships and private groups. The sensitivity of the data often prevents the data owner from publishing it. Network graph researchers analyze the graph for graph strength, nodes and types of groups. These traces represent a social network where the entities are internet hosts and the existence of communication between hosts constitutes a relationship. However network traces sensitive information because it is often possible to associate individuals with the hosts they use, and because traces contain information about web sites visited, and time stamps which indicate periods of activity. The challenges in protecting network trace data are being actively addressed by the research community.

The objective of the data owner is to publish the data in such a way that permits useful analysis yet avoids disclosing sensitive information. Because network analysis can be performed in the absence of entity identifiers, the data owner first replaces identifying attributes with synthetic identifiers. We refer to this procedure as naive anonymization. It is a common practice in many domains, and it is often implemented by simply encrypting identifiers. Presumably, it protects sensitive information because it breaks the association between the sensitive data and real-world individuals. Before publishing all these data for analysis, data mining, and other purposes, it is necessary to ensure that the published data will not contain any private information.

Most existing work on privacy in data publishing has focused on tabular data, where each record represents a separate entity, and an individual may be re-identified by matching the individual's publicly known attributes with the attributes of the anonymized table. Anonymization techniques for tabular data do not apply to networked data because they fail to account for the interconnectedness of the entities. It is not well-understood how publishing a network threatens privacy; initial investigations, including anonymization algorithms, are just emerging. Formally, we model a network as an undirected graph $G=(V,E)$. The naive anonymization of G is an isomorphic graph, $G_a = (V_a, Ea)$, defined by a random bijection function $f: V \rightarrow Va$.

Figure 1 shows a small network along with its naive anonymization. The anonymization mapping f, is a random, secret mapping. Naive anonymization prevents re-identification when the adversary has no information about individuals in the original graph. To assess the risk of re identification, we assume each element of the candidate set is equally likely and use the size of the candidate set as a measure of resistance to re-identification. Since f is random, in the absence of other information, any node in G_a could correspond to the target node x. Thus, given an uninformed adversary, each individual has the same risk of re-identification.



(a)

(b)

Gnanam	3
Kesaven	1
Jayanthi	5
Anbu	4
Munusamy	6
Priya	2

(c)

Figure 1: (a) Social Network (G), (b)Naïve Anonymized Network(G'),(c)Mapping function (f)

However, in practice the adversary may have access to external information about the entities in the graph and their relationships. This information may be available through a public source beyond the control of the data owner, or may be obtained by the adversary's malicious actions. For example, for the graph in Figure 1, the adversary might know that "Munusamy has three or more neighbors," or that "Gnanam is connected to at least two nodes, each with degree 2." Such information allows the adversary to reduce the set of candidates in the anonymized graph for each of the targeted individuals. Although an adversary may also have information about the attributes of nodes, the focus of this paper is structural re-identification, where the adversary's information is about graph structure. Re-identification with attribute knowledge has been well-studied, as have techniques for resisting it. More importantly, many network analyses are concerned exclusively with structural properties of the graph; therefore safely publishing an unlabeled network is a legitimate goal.

This above example shows that a naive privacy preserving published network is still susceptible to these structural attacks. However, these suffer from the following limitations.

1) The previously research papers tried about only on anonymization and only on any single type of attack. Probably an adversary may use any type of attack and we are not sure about his technique of attack on structure. Also it is not sure that an adversary will use only one type of attack. So, the researches has to tackle simultaneous multiple attacks.

2) Since the released network only contains a summary of structural information about the original network, users have to generate some random sample instances of the released network for further analysis. Introducing uncertainty in the released network can enable handling of structural attack. This is the only main solution to handle multiple attacks. By introducing uncertainty it makes an adversary to analyze the network structure.

3) Existing methods do not consider dynamic releases. This is important in evolutionary networks and dynamic social network analysis. For example, given a series of online trading networks, such as eBay, based on community evolution in these networks, we can predict the trend of consumers' purchasing behavior. These applications require republishing data periodically to support dynamic analysis. However, all existing privacy-preserving network publication methods consider only "one-time" release. Even though each released network G_t^* at time T_t can guarantee privacy individually, an adversary can still identify the target with a high probability by collecting the information from multiple releases.

II. Related Work

Before formally describing adversary we consider the practical properties of adversary knowledge that motivate our definitions. We also explain how structural similarity in a graph can protect against structural attack.

*2.1 **Knowledge Acquisition in Practice***

Accurately modeling adversary knowledge is a key for understanding the vulnerabilities of naively-anonymized networks, and for developing new anonymization strategies. External information about a published social network may be acquired through malicious actions by the adversary or from public information sources. In addition, a participant in the network, with some innate knowledge of entities and their relationships, may be acting as an adversary in an attempt to uncover unknown information. A legitimate privacy objective in some settings is to publish a network in which participating individuals cannot re-identify themselves.

Our goal is to develop parameterized and conservative models of external information that capture the power of a range of adversaries, and to then study the threats to anonymity that result. One of our guiding principles is that adversary knowledge about a targeted individual tends to be local to the targeted node, with more powerful adversaries capable of exploring the neighborhood around a node with increasing diameter. For the participant-adversary, whose knowledge is based on their participation in the network, existing research about institutional communication networks suggests that there is a horizon of awareness of about distance two around most individuals. We formalize the external information available to an adversary through a set of knowledge queries described in the next section. Each knowledge query is parameterized by the radius around the targeted individual which it describes.

We also consider the impact of hubs, which are connected nodes observed in many networked data sets. In a Web graph, a hub may be a highly visited website. In a graph of email connections, hubs often represent influential individuals. Because hubs are often outliers in a graph's degree distribution, the true identity of hub nodes is often apparent in a naively-anonymized graph. In addition, an individual's connections to hubs may be publicly known or easily deduced. We consider attackers who use hub connections as a structural fingerprint to re-identify nodes.

Our assumption throughout the present work is that external information sources are accurate, but not necessarily complete. Accuracy means that when an adversary learns facts about a named individual, those facts are true of the original graph. However, we distinguish between a closed-world adversary, in which absent facts are false, and an open-world adversary in which absent facts are simply unknown. For example, when a closed-world adversary learns that Munusamy has three neighbors, he also learns that Munusamy has no more than three neighbors. An open-world adversary would learn only that Munusamy has at least three neighbors. Hub fingerprints have an analogous open- and closed-world interpretation.

In practice, an adversary may acquire knowledge that is complete. For example, an attacker who acquires the address book for a targeted individual would learn a complete list of their neighbors in an email communication network.

As we would expect, closed-world adversaries are significantly more powerful. However, in many settings, the adversary cannot be certain that their information is complete and must assume an open world. We believe both closed- and open-world variants of adversary knowledge are important.

2.2 *Anonymity through Structural Similarity*

Naturally, nodes that seem structurally similar may be impossible to differentiate to an adversary, in spite of external information. A strong form of structural similarity between nodes is automorphic equivalence. Two nodes x, y are automorphically equivalent if there exists an isomorphism from the graph onto itself that maps x to y.

EXAMPLE 2.1. Priya and Munusamy are automorphically equivalent nodes in the graph of Figure 1. Munusamy and Gnanam are not automorphically equivalent: the subgraph around Munusamy is different from the subgraph around Gnanam and no isomorphism proving automorphically equivalence is possible.

An Automorphically equivalence structure induces a partitioning on V into sets whose members have indistinguishable structural properties. It follows that an adversary even with comprehensive knowledge of a target node's structural position cannot identify an individual beyond the set of entities to which it is automorphically equivalent. We say that these nodes are structurally indistinguishable and observe that nodes in the graph achieve anonymity by being "hidden in the crowd" of its automorphically class members. Some special graphs have large automorphically equivalence classes. For example, in a complete graph, or in a graph which forms a ring, all nodes are automorphically equivalent. But in most graphs we expect to find small automorphism classes, likely to be insufficient for protection against re-identification. Though automorphism classes may be small in real networks, automorphically equivalence is an extremely strong notion of structural similarity. In order to distinguish two nodes in different automorphically equivalence classes, it may be necessary to use complete information about their positions in the graph. For example, for a weaker adversary, who only knows the degree of targeted nodes in the graph, Munusamy and Gnanam are indistinguishable. Thus we must consider the distinguishability of nodes to realistic adversaries with limited external information.

2.3 *Targets of Protection*

Privacy preservation is about the protection of sensitive information. This may concern with nodes, edges, relationship between the nodes, and network structure. An adversary may have background knowledge on network structure. From the examples of real datasets we identify two main types of sensitive information that a user may want to keep private and which may be under attack in a social network.

2.3.1 Node Info:

The first type of target to protect is node and we can it as *Node Info*.

For example, the emails sent by an individual in the Enron dataset can be highly sensitive since some of the emails have been written only for private recipients and should not be allowed to be linked to any individual.

We assume that any identifying information such as names will first be removed from Node Info, so that the content of Node Info does not help the identification of its owner.

2.3.2 Link Info:

The second type, which we call *Link Info*, is the information about the relationships among the individuals, which may also be considered sensitive. In this case, the adversary may target at two different individuals in the network and try to find out if they are connected by some path.

We aim to provide sufficient protection for both Node Info and Link Info. We should point out that the linkage of an individual to a node in the published graph itself does not disclose any sensitive information for the Node Info target, because if we separate the publishing of the Node Info from that of the node, then attacks of the first type will not be possible.

III. K-AUTO RESTRUCTURE

In order to guarantee privacy from any structural attack, we propose the following concept.

Definition 3.1. k-*Autorestructured Network. Given a network G, (a) if there exist k-1 autorestructure functions* F_a *(a=1,...,k-1) in G, and (b) for each vertex v in G,* F_{a1} *(v)* $\neq F_{a2}$ *(v) (1≤ a1≠a2 ≤ k -1), then G is called a k-restructured network.*

Obviously, if G is a k-autorestructured network, for any vertex v in G, we cannot distinguish *v* from its *k-1* symmetric vertices based on any structural information. Thus, an adversary cannot identify *v* from G with a probability higher than$\frac{1}{K}$. Therefore, the problem that we want to solve in this paper is defined as follows:

Definition 3.2. Given an original network G, find a network G, where G is a sub-graph of G* and G* is a k-autorestructured network. G* is published as G's anonymized version.*

3.1 ***Node Anonymization***

We assume that the nodes have been anonymized with one of the techniques introduced for single table data. For example, the nodes could be k-anonymized using t-closeness. This anonymization provides a clustering of the nodes into m equivalence classes (C_1, . . . ,C_m) such that each node is indistinguishable in its quasi-identifying attributes from some minimum number of other nodes. We use the following notation $C(v_i) = C_k$ to specify that a node v_i belongs to equivalence class C_k. The anonymization of nodes creates equivalent classes of nodes. Note, however, that these equivalent classes are based on node attributes only, and inside each equivalence class, there may be nodes with different identifying structural properties and edges.

ALGORITHM

```
Input:G=(V,E^1,...,E^s)
Output: G' = (V 0,E10,... ,Ek0)
V'=anonymize-nodes(V)
for t=1 to k do
E^t' = Et
end for
```

3.2 Edge Anonymization

The first edge anonymization option is to only remove the sensitive edges, leaving all other observational edges intact. In our running example, we remove the friendship relationships, since they are the sensitive relationships, but we leave intact the information about students taking classes together and being members of the same research group. Since the relational observations remain in the graph, this anonymization technique should have a high utility. But it is likely to have low privacy preservation.

3.3 Partial-edge removal

Another anonymization option is to remove some portion of the relational observations. We could either remove a particular type of observation which contributes to the overall likelihood of a sensitive relationship, or remove a certain percentage of observations that meet some pre-specified criteria. This partial edge removal process should increase the privacy preservation and reduce the utility of the data as compared to the previous method. Removing observations should reduce the number of node pairs with highly likely sensitive relationships but it does not remove them completely. For those pairs of nodes, private information may be disclosed.

3.4. ***Cluster-edge anonymization***

In the above approaches, while the nodes had been anonymized, the number of nodes in the graph was still the same, and the edges were essentially between copies of the anonymized nodes. Another approach is to collapse the anonymized nodes into a single node for each cluster, and then consider which edges to include in the collapsed graph.

Definition 3.4 **k-*Autorestructure clustered social network*:** *An anonymized social network G* = (u,v), where u = {C_1,C_2, ... , C_v}, and C_j = [(|c_j|, |Ec_j|)], j = 1, ..., v is k-anonymous iff |cj| ≥ k for all j = 1, ..., v.*

The algorithm used in the anonymization process, called the *SaNGreeA* (Social Network Greedy Anonymization) algorithm, performs a greedy clustering processing of an initial social network in order to generate a *k*-anonymous clustered social network. In this algorithm the nodes that are more similar in terms of their neighborhood structure are clustered together using a greedy approach. To do so, a measure that quantifies the extent to which the neighborhoods of two nodes are similar with each other is used.

IV. Graph-Based Privacy Attacks

According to Li et. al., there are two types of privacy attacks in data: identity disclosure and attribute disclosure. In graph data, there is a third type of attack: link re-identification.

Identity disclosure occurs when the adversary is able to determine the mapping from an anonymized record to a specific real-world entity. Attribute disclosure occurs when the adversary is able to infer the attributes of a real world entity more accurately than it would be possible before the data release. Identity disclosure often leads to attribute disclosure. Both identity disclosure and attribute disclosure have been studied widely in the privacy community.

Rather than focus on these two kinds of attack, the focus of our paper is on link re-identification. Link re-identification is the problem of inferring that two entities participate in a particular type of sensitive relationship or communication. Sensitive conclusions are more general statements that an adversary can make about the data, and can involve node, edge and structural information. These conclusions can be the results of aggregate queries. For example, in a database describing medical data informal about company employees, finding that almost all people who work for a particular company have a drinking problem may be undesirable. Depending on the representation of the data, this can be revealed by using both the node attributes and the co-worker relationship.

V. Link Re-Identification Attacks

The extent of a privacy breach is often determined by data domain knowledge of the adversary. The domain knowledge can influence accurate inference in subtle ways. The goal of the adversary is to determine whether a sensitive relationship exists. There are different types of information that can be used to infer a sensitive relationship: node attributes, edge existence, and structural properties. Based on the domain knowledge of the adversary, she can construct rules for finding likely sensitive relationships. In this work, we assume that the adversary has an accurate probabilistic model for link prediction, which we will describe below. In our running example, the sensitive friendship link may be re-identified based on node attributes, edge existence or structural properties. For example, consider two student nodes containing a boolean attribute "Talkative." Two nodes that both have it set to "true" may be more likely to be friends than two nodes that both have it set to "false." This inference is based on node attributes. An example of re-identification based on edge existence is two students in the same research group who are more likely to be friends compared to if they are in different research groups. A re-identification that is based on a structural property such as node degree would say that two students are more likely to be friends if they are likely to correspond to high degree nodes in the graph. A more complex observation is one which uses the result of an inferred relationship. For example, if each of two students is highly likely to be a friend with a third person based on other observations, and then the two students are more likely to be friends too.

VI. Link Re-Identification In Anonymized Data

In the first two types of link anonymization, the noisy-or model can be used directly to compute the probability of a sensitive edge. In the other two cases, one has to consider the probability that an observed edge exists between two nodes, and apply the noisy-or.

6.1 Link re-identification in cluster-edge anonymization

In the case of keeping edges between equivalence classes, the probability of an observation existing between two nodes is not given and it needs to be estimated. The noisy-or function will need to take into consideration the probability associated with each observation in order to compute the likelihood of a sensitive relationship. When the number of relationships of each type between two equivalence classes is given, the distribution is not uniform, and the probability of an observation *P(o)=P(observation(vi, vj))* existing between two students can be computed directly from the counts of relationships between their equivalence classes. *P(classmates(vi,vj,c))* expresses the probability that there exists a class edge between any two students vi and vj from two equivalence classes $C(v_i)$ *and* $C(v_j)$, i.e., the students take a course c together. It is equal to the number of possible student pairs from the two equivalence classes who take a course together $classmates(C(v_i),C(v_j))$ as a fraction of the number of possible relationships in the graph $|V|^2$.

6.2 Link re-identification in cluster-edge anonymization with constraints

In the constrained cluster-edge anonymization approach, the number of relationships between equivalence classes is not given. Therefore, the probability of an observation existing between any two edges has to be taken into account in the noisy-or model. To estimate this probability, an adversary can assume a uniform distribution, meaning that the probability of an observation existing between any two edges is the same for all edges in the graph. This estimate is worse than the cluster-edge anonymization method. Using the constraints on the data, it is possible to get estimates of this probability. For example, if it is known that there are 50 pairs of students who take courses together, and there are 100 possible pairs, then the probability of any two students taking any class c together is *P(classmates(vi, vj , c))=0.5*. If the adversary knows the number of offered courses c, the number of courses per person n, the number of students s = |V|, and assumes that all courses have the same number of people $p = \frac{s*n}{c}$ then the number of possible pairs who take courses together can be calculated as $n*(p - 1)$. This number can be used to compute in a manner similar to the cluster-edge anonymization method $P(classmates(v_i,v_j, c))=n*(p-1)$ |V|2.

One can also use an expected value of any two-node relationship to be sensitive by looking at the likelihood distribution of all relationships. However, we found that this does not measure privacy well because an adversary is more interested in the highly likely relationships.

An observation probability shows the percentage of edges between two nodes from two different equivalence classes that contain the observation. For example, if the two equivalence classes have exactly 10 nodes each, and the observation exists for 30 of the two-node edges, then the edge probability is $P(observation(v_i, v_j))=0.3$ where $observation(v_i, v_j)$ is either $classmates(v_i,v_j,c)$, or *group mates*(v_i,v_j,g) for any c and g. This increases the utility of the data as compared to the case when no probabilities are included, but it can also decrease the privacy preservation. An exception is the case when observations between equivalence classes have exactly the same distribution as the overall uniform distribution.

VII. Experiments

We study the above illustrated structural properties on the original and de-anonymized versions of several real and synthetic datasets. These datasets are described next.

The Enron dataset is a network of e-mail exchanges available online. A node in this network represents an email address. An edge exists between two nodes if at least one e-mail was sent from one node to the other node from that edge. This network has 36,692 nodes and 183,831 edges.

The Scale Free dataset is an undirected network generated based on the scale free model. This approach models real world social networks that follow a power-law degree distribution. We generated this dataset using following initial parameters: the number of nodes = 10,000, average degree of nodes = 33.The generated graph has a significant number of multiple edges (more than 60,000) which are eliminated in a post-processing step. This final scale free network that we used in our experiments has 10,000 nodes and 100,657 edges.

The last dataset, labeled RMAT, is based on the R-MAT model. We implemented an R-MAT graph generator that takes the number of nodes (*n*), the average node degree (*avg_deg*), and four probabilities as input parameters. The location of each edge is determined based on a recursive algorithm that divides the adjacency matrix into 4 equal-sized partitions and the edge location is probabilistically selected in one of the 4 partitions, based on the four probability parameters (we used the values 0.45, 0.15, 0.15, and 0.25 for RMAT dataset generation). Once a partition is decided, it is again divided into four sub-partitions until there will be only one cell from the adjacency matrix left in the partition. If this cell has value 1, this procedure is repeated from the beginning. This approach also models real-world graphs that follow power-law degree distributions. We start from the initial social networks (Enron, Scale Free, and RMAT) previously described. First, the initial social networks are anonymized into *k*-anonymous clustered social networks as described in Section 2. For each dataset we used the following values for *k*: 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 25, and 50.

Second, from each k-anonymous clustered social network ten possible de-anonymized social networks are generated. In this process we generate edges randomly within each cluster until the number of generated edges is equal to the number of edges recorded in the super-node description. This process continues with the generation of edges between super nodes. These edges are also generated randomly until the number of generated edges is equal with the number of edges that describes the corresponding super-edge between the two super-nodes. This process guarantees that each generated "de-anonymized" social network will have the same number of nodes and edges as the original network. We decided to generate ten networks for each anonymized social network to avoid any possible outliers. In Steps 3 and 4, we compute the structural properties values for the original and de-anonymized social networks. Since we generated ten social networks for each anonymized network, we report the average for each structural property. Last, we compare the structural properties values measured for the original social network with the ones obtained from the anonymized networks. The results are shown in Figures 2. The vertical axis shows the probability that an adversary can perform structural attack and find the network structure.. The reason we chose to report the values or the ratio is due to the fact that the values can be very different between the five considered datasets and the representation of all the values is difficult to include in one chart. Horizontal axis shows nodes from various data source. On the other hand, for some structural properties (such as diameter) reporting the values provides more information. The Figure 2 shows how the probability gets down as the network is refine at various steps.

7.1 Dataset

This dataset consists of 'circles' (or 'friends lists') from Facebook. Facebook data was collected from survey participants using this Facebook app. The dataset includes node features (profiles), circles, and ego networks.

Facebook data has been anonymized by replacing the Facebook-internal ids for each user with a new value. Also, while feature vectors from this dataset have been provided, the interpretation of those features has been obscured. For instance, where the original dataset may have contained a feature "political=Democratic Party", the new data would simply contain "political=anonymized feature 1". Thus, using the anonymized data it is possible to determine whether two users have the same political affiliations, but not what their individual political affiliations represent.

Dataset statistics	
Nodes	4039
Edges	88234
Nodes in largest WCC	4039 (1.000)
Edges in largest WCC	88234 (1.000)
Nodes in largest SCC	4039 (1.000)
Edges in largest SCC	88234 (1.000)
Average clustering coefficient	0.6055
Number of triangles	1612010
Fraction of closed triangles	0.2647
Diameter (longest shortest path)	8
90-percentile effective diameter	4.7

7.2 ***Data Generator***

The data generator creates data according to the data model described in Section 3. The input to the data generator includes: the number of nodes, maximum number of nodes which can participate in a relationship (e.g., the maximum number of students taking the same class), the maximum number of relationships that each student can have with any other student (e.g., maximum number of classes that a student can take). For all observation types, the probability of two nodes exhibiting a sensitive relationship given the observation type is given and the leak probability, the probability of two nodes exhibiting a sensitive relationship due to unobserved causes.

For the concrete example, the data generator starts by creating a set of students, a set of classes, and a set of research groups. There are constraints on how many classes each student takes, and on how many research groups each student belongs. There are also constraints on the maximum number of students per class and on the maximum number of students per group. For each student, the generator picks random classes to enroll into up to the maximum number of classes per student possible. Similarly, each student is assigned to a random research group.

The nodes in the data graph represent students. There is a class mates edge connecting two students for each class they take together, and there is groupmates edge if they belong to the same research group. These pieces of information represent observations indicating that two students may be friends, i.e., that they may exhibit a sensitive relationship. The ground truth is generated by computing the probability of a friendship between each two students using the noisy-or model, and assigning the friendship a true value with a probability equal to that likelihood.

The parameters given to the data generator can be varied. We would like to explore graphs which vary in their density; therefore we allow the number of lasses and research groups to vary while fixing the number of nodes/students to 100. The constraints on the data are that each student takes two classes, and belongs to one research group. Also, a class can have no more than 25 people, and a group can have no more than 15. We picked probabilities which make sense in the domain. The prior probability of two students knowing each other is $P(friends(v_i, v_j))=0.2$. It is relatively high because the students are from the same department. The probability that two students know each other if they are in the same class c is $P(friends(v_i,v_j)\text{-}classmates(v_i, v_j,c))=0.4$. The probability that two students know each other if they are in the same research group is $P(friends(v_i,v_j)\text{-}groupmates(v_i,v_j,c))=0.6$.

7.3 ***Evaluating privacy preservation in anonymized data***

We begin by studying the privacy preservation in the data that results from each of the anonymization techniques. In particular, we study the number of correctly identified sensitive relationships for the following anonymization functions:

i. When the anonymization function leaves the edges between nodes intact (4.2),
ii. When it removes 50% of the observations chosen at random (4.2),
iii. When it leaves edges between node equivalence classes in the cluster-edge anonymization (4.2), and
iv. When it leaves edges between node equivalence classes with a constrained number of observations (4.2). For the last two, each node is assigned randomly to an equivalence class. We vary k, the number of nodes in each equivalence class, and show the results for k = 2 and k = 6 because they exhibit the tendencies of varying k well.

The data was generated with the default parameters, varying the number of classes and the number of research groups between 10 and 30. A graph, in which there are 10 research groups and 10 classes, is very dense, and a graph at the other extreme with 30 research groups and 30 classes is very sparse. To account for the randomness in the generated graph, we ran the experiments on 100 generated graphs, and present the average performance. Note that when using the default data parameters, the maximum possible likelihood for their friendship is 0.89.

We measure the probability of attack on network structure again nodes from various nodes.

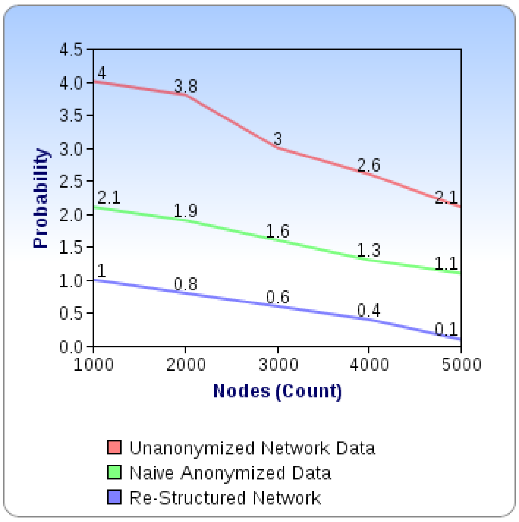


Figure 2: Nodes Vs Probability of Structural Attack

7.4 Results

Figure 2 shows a comparison between the numbers of sensitive relationships inferred after each of our proposed anonymization technique has been applied. It shows that at higher thresholds (0.6 and 0.8) keeping all the edges between node equivalence classes preserves privacy much better than deleting 50% of the two-node edges, while having higher utility. As expected, for lower k, the privacy preservation is lower: the number of revealed relationships is higher in the data anonymized with the cluster-edge method. In the data anonymized with the cluster-edge method with constraints, varying k yielded to the same results, which is why the graphs of k = 2 overlap with the graphs, in which k = 6.

VIII. CONCLUSION

We have focused on what we believe to be one of the most basic and distinctive challenges for protecting privacy in network data sets understanding the extent to which graph structure acts as an identifier. We have formalized classes of adversary knowledge and evaluated their impact on real networks as well as models of random graphs. We proposed anonymizing a graph by generalizing it: partitioning the nodes and summarizing the graph at the partition level. We show that a wide range of important graph analyses can be performed accurately on a generalized graph while protecting against re-identification risk.

REFERENCES

[1] V. Rastogi, S. Hong, and D. Suciu. *The boundary between privacy and utility in data publishing*. In VLDB, 2007.

[2] S. Russell and P. Norvig. AI: A Modern Approach. 2003.

[3] L. Singh and J. Zhan. *Measuring topological anonymity in social networks.* In Intl. Conf. on Granular Computing, 2007.

[4] L. Sweeney. *k-anonymity: a model for protecting privacy.* Journ. of Uncertainty, Fuzziness, and KB Systems, 2002.

[5] D.-W. Wang, C.-J. Liau, and T.-S. Hsu. *Privacy protection in social network data disclosure based on granular computing.* In International Conference on Fuzzy Systems, 2006.

[6] D. J. Watts and S. H. Strogatz. *Collective dynamics of 'small-world' networks.* Nature, 393:440–442, 1998.

[7] D. B. West. *Introduction to Graph Theory*. August 2000.

[8] X. Ying and X. Wu. *Randomizing social networks: a spectrum preserving approach.* In SIAM Conf. on Data Mining, 2007.

[9] L. Getoor and C. P. Diehl. *Link mining: a survey*. SIGKDD Explor. Newsl.,7(2):3–12, December 2005.

[10] M. Hay, G. Miklau, D. Jensen, P. Weis, and S. Srivastava. *Anonymizing social networks,* March 2007.

[11] N. Li, T. Li, and S. Venkatasubramanian. *t-closeness: Privacy beyond k-anonymity and l-diversity.* In IEEE 23rd International Conference on Data Engineering, pages 106–115, April 2007.

[12] A. Machanavajjhala, J. Gehrke, D. Kifer, and M.Venkitasubramaniam.*l-diversity: Privacy beyond k-anonymity.* In 22nd IEEE International Conference on Data Engineering, 2006.

[13] G. Miklau and D. Suciu. *A formal analysis of information disclosure in data exchange.* In ACM Conference on Management of Data (SIGMOD), pages 575–586, 2004.

[14] M. E. Nergiz, M. Atzori, and C. Clifton. *Hiding the presence of individuals from shared databases.* In 26th ACM SIGMOD International Conference on Management of Data, June 2007.

[15] C. Dwork, F. McSherry, K. Nissim, and A. Smith. *Calibrating noise to sensitivity in private data analysis.* In TCC, 2006.