IISRC International Industrial Scientific Research Corporation
International Conference on Industrial Scientific Research Engineering
www.worldconferences.org

Protecting Private Information in Social Networks

PENCHALAPRASAD KAKIVAI[1], S.PRATHAP[2]
[1]PG Scholar, Dept of CS, AITS, Tirupati, AP, India, Email: prasad5232@gmail.com.
[2]Asst Prof, Dept of CSE, AITS, Tirupati, AP, India, Email: prathapforall@gmail.com.

Abstract: The popularity of social network services is increasing enormously. Generally in social networks different people of different community communicate with each other through their neighbors or similar community members. The adversary always tries to access the information about the users. In previous cases, social networks provide security to user identities. But Users wants privacy to their sensitive information like address, phone number etc. Now it becomes trend to provide security to features of social networks. For easy to understand the social networks are represented as graphs. Here users are called nodes, communication between them is called link and features are called labels. Labels are two types. They are Sensitive and non-sensitive labels. Also we developed a privacy protection algorithm this allows for graph data to be published in a form such that an adversary who possess information about nodes neighbor cannot safely infer its identity and its sensitive label. In order to achieve this, the algorithm transformed the original graph into the graph in which the nodes are sufficiently indistinguishable. Specifically in this paper we proposed a scheme that not only prevents the disclosure of user's identity but also the disclosure of selected features in user's profile. Also an individual user can select the features which he wishes to conceal. Here this algorithm provides stronger privacy guarantee than compared to previous algorithms.

Keywords: Social Networks, Sensitive Information, Access Information, Privacy.

I. INTRODUCTION

With the rapid growth of social networks, such as Facebook and Linkedin, more and more researchers found that it is a great opportunity to obtain useful information from these social network data, such as the user behavior, community growth, disease spreading, etc. However, it is paramount that published social network data should not reveal private information of individuals to others. Thus, how to protect individual's privacy and at the same time preserve the utility of social network data becomes a challenging topic. In this paper, a graph model where each vertex in the graph is associated with a sensitive label. Recently, much work has been done on anonymizing tabular micro data. A variety of privacy models as well as anonymization algorithms have been developed. In tabular micro data, some of the non-sensitive attributes, called quasi identifiers, can be used to re-identify individuals and their sensitive attributes. When publishing social network data, graph structures are also published with corresponding social relationships. As a result, it may be exploited as a new means to compromise privacy.

A structure attack refers to an attack that uses the structure information, such as the degree and the sub graph of a node, to identify the node. To prevent structure attacks, a published graph should satisfy k-anonymity. The goal is to publish a social graph, which always has at least k candidates in different attack scenarios in order to protect privacy. Previous work in this direction that defined a k-degree anonymity model to prevent degree attacks (Attacks use the degree of a node). A graph is k-degree anonymous if and only if for any node in this graph, there exist at least k - 1 other node with the same degree. If an adversary knows that one person has three friends in the graph, he can immediately know that node 2 is that person and the related attributes of node 2 are revealed. K-degree anonymity can be used to prevent such structure attacks. However, in many applications, a social network where each node has sensitive attributes should be published. For example, a graph may contain the user salaries which are sensitive. In this case, K-degree alone is not sufficient to prevent the inference of sensitive attributes of individuals.

A graph that satisfies 2-degree anonymity but node labels is not considered. In it, nodes 2 and 3 have the same degree 3, but they both have the label "80K." If an attacker knows someone has three friends in the social network, he can conclude that this person's salary is 80K without exactly reidentifying the node. Therefore, when sensitive labels are considered, the l diversity should be adopted for graphs. Again, the l-diversity concept here has the same meaning as that defined over tabular data. For example, if the distinct l-diversity, for the nodes with the same degree, their associated sensitive labels must have l distinct values. For each distinct degree appearing in this graph, there exist at least two nodes. Moreover, for those nodes with the same degree, they contain at least two distinct sensitive labels. Thus, the attacker cannot re-identify a node or find the

node-label relation with degree knowledge. In this paper, select the degree-attack, one of the popular attacks methods, to show design mechanisms to protect both identities and sensitive labels. With respect to other types of attacks, such as sub graph query attacks or hub node query attacks, that the key ideas proposed in this work can be adopted to handle them as well, though more complicated extensions may be needed.

Current approaches for protecting graph privacy can be classified into two categories: clustering and edge editing. Clustering is to merge a sub graph to one super node, which is unsuitable for sensitive labeled graphs, since when a group of nodes are merged into one super node the node-label relations have been lost. Edge-editing methods keep the nodes in the original graph unchanged and only add/delete/swap edges. For example, to protect privacy, and convert it to satisfy 3-degree anonymous and 3-diversity by adding edges (fig 1). However, edge editing may largely destroy the properties of a graph. The edge editing method sometimes may change the distance properties substantially by connecting two faraway nodes together or deleting the bridge link between two communities. In the distance between nodes 6 and 12 is changed from 5 to 1 hop.

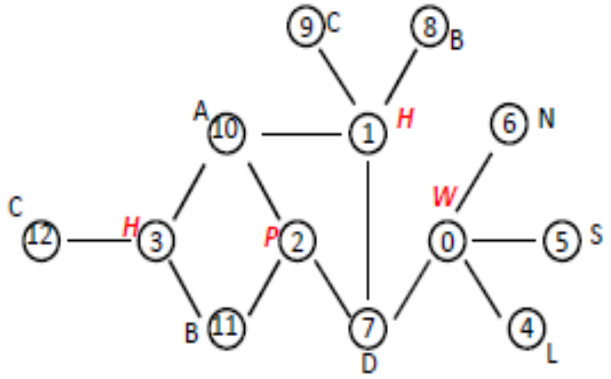

Fig.1. Example of the labeled graph representing a social network.

This phenomenon is not preferred. Mining over these data might get the wrong conclusion about how the salaries are distributed in the society. Therefore, solely relying on edge editing may not be a good solution to preserve data utility. To address this issue, A novel idea to preserve important graph properties, such as distances between nodes by adding certain "noise" nodes into a graph. This idea is based on the following key observation. Most social networks satisfy the Power Law distribution i.e., there exist a large number of low degree vertices in the graph which could be used to hide added noise nodes from being re-identified. By carefully inserting noise nodes, some graph properties could be better preserved than a pure edge-editing method. The distances between the original nodes are mostly preserved. Our privacy preserving goal is to prevent an attacker from re-identifying a user and finding the fact that a certain user has a specific sensitive value. To achieve this goal, to define a k-degree-l-diversity (KDLD) model for safely publishing a labeled graph, and then develop corresponding graph anonymization algorithms with the least distortion to the properties of the original graph, such as degrees and distances between nodes. Analytical results to show the relationship between the number of noise nodes added and their impacts on an important graph property. Further conduct comprehensive experiments for both distinct l-diversity and recursive (c, l)-diversity to show our technique's effectiveness.

II. RELATED WORK

To secure sensitive Information in social network data anonymization using k-degree-l-diversity anonymity model.

A. Objective of project

- Privacy is one of the major concerns when publishing or sharing social network data for social science research and business analysis.
- The label-node relationship is not well protected by pure structure anonymization methods.
- K-degree l-diversity anonymity model that considers the protection of structural information as well as sensitive labels of individuals.
- Adding noise nodes into the original graph with the consideration of introducing the least distortion to graph properties.

B. Existing system

The current trend in the Social Network it not giving the privacy about user profile views. The method of data sharing or (Posting) has taking more time and not under the certain condition of displaying sensitive and non sensitive data.

C. Edge-Editing –Based Model

The edge editing- based model is to add or delete edges to make the graph satisfy certain properties according to the privacy requirements. Most edge-editing-based graph protection models implement k-anonymity of nodes on different background knowledge of the attacker. Liu and Terzi defined and implemented k-degree-anonymous model on network structure that is for published network, for any node, there exists at least other k-1 nodes have the same degree as this node. Zhou and Pei considered k-neighborhood anonymous model: for every node, there exist at least other k-1 nodes sharing isomorphic neighborhoods.

D. Clustering-Based Model

Clustering-based model is to cluster "similar" nodes together to form super nodes. Each super node represents several nodes which are also called a "cluster." Then, the links between nodes are represented as the edges between super nodes which is called "super edges." Each super edge may represent more than one edge in the original graph. The graph that only contains super nodes and super edges are called as clustered graph.

E. Drawbacks of existing

- There is no way to publish the Non sensitive data to all in social Network.

- It's not providing privacy about user profiles.
- Some mechanisms that prevent both inadvertent private information leakage and attacks by malicious adversaries.

III. PROBLEM DEFINITION

We model a network as G (V, E, L^s, L, Γ), where V is a set of nodes, E is s set of edges, L^s is a set of sensitive labels, and L is a set of non-sensitive labels. Maps nodes to their labels, Γ: V → $L^s \cup L$. Then we propose a privacy model, l-sensitive-label-diversity; in this model, we treat node labels both as part of an adversary's background knowledge, and as sensitive information that has to be protected. These concepts are clarified by the following definitions:

Definition 1: The neighborhood information of node v comprises the degree of v and the labels of v's neighbors.

Definition 2: (L-sensitive-label-diversity) for each node v that associates with a sensitive label, there must be at least L-1 other nodes with the same neighborhood information, but attached with different sensitive labels.

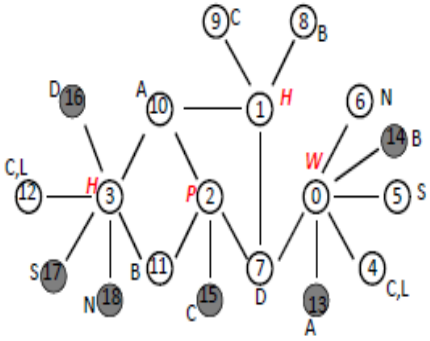

Fig 2: Privacy attaining network example.

In Example 1, nodes 0, 1, 2, and 3 have sensitive labels. The neighborhood information of node 0, includes its degree, which is 4, and the labels on nodes 4, 5, 6, and 7, which are L, S, N, and D, respectively. For node 2, the neighborhood information includes degree 3 and the labels on nodes 7, 10, and 11, which are D, A, and B. The graph in Figure 2 satisfies 2-sensitive-label-diversity, that is because, in this graph, nodes 0 and 3 are indistinguishable, having six neighbors with label A, B, {C, L}, D, S, N separately; likewise, nodes 1 and 2 are indistinguishable, as they both have four neighbors with labels A, B, C, D separately.

IV. ALGORITHM USED

The main objective of the algorithms that we propose is to make suitable grouping of nodes, and appropriate modification of neighbor's labels of nodes of each group to satisfy the l-sensitive-label-diversity requirement. We want to group nodes with as similar neighborhood information as possible so that we can change as few labels as possible and add as few noisy nodes as possible. We propose an algorithm, Global-similarity-based Indirect Noise Node (GINN) that does not attempt to heuristically prune the similarity computation as the other two algorithms, Direct Noisy Node Algorithm (DNN) and Indirect Noisy Node Algorithm (INN) do. Algorithm DNN and INN, which we devise first, sort nodes by degree and compare neighborhood information of nodes with similar degree.

A. Algorithm GINN

The algorithm starts out with group formation, during which all nodes that have not yet been grouped are taken into consideration, in clustering-like fashion. In the first run, two nodes with the maximum similarity of their neighborhood labels are grouped together. Their neighbor labels are modified to be the same immediately so that nodes in one group always have the same neighbor labels. For two nodes, v1 with neighborhood label set (LS_{v1}), and v2 with neighborhood label set (LS_{v2}), we calculate neighborhood label similarity (NLS) as follows:

$$NLS(v1, v2) = |(LS_{v1}) \cap (LS_{v2})| / |(LS_{v1}) \cup (LS_{v2})| \quad (1)$$

Larger value indicates larger similarity of the two neighborhoods. Then nodes having the maximum similarity with any node in the group are clustered into the group till the group has ` nodes with different sensitive labels. Thereafter, the algorithm proceeds to create the next group. If fewer than ` nodes are left after the last group's formation, these remainder nodes are clustered into existing groups according to the similarities between nodes and groups. After having formed these groups, we need to ensure that each group's members are indistinguishable in terms of neighborhood information. Thus, neighborhood labels are modified after every grouping operation, so that labels of nodes can be accordingly updated immediately for the next grouping operation. This modification process ensures that all nodes in a group have the same neighborhood information.

The objective is achieved by a series of modification operations. To modify graph with as low information loss as possible, we devise three modification operations: label union, edge insertion and noise node addition. Label union and edge insertion among nearby nodes are preferred to node addition, as they incur less alteration to the overall graph structure. Edge insertion is to complement for both a missing label and insufficient degree value. A node is linked to an existing nearby (two-hop away) node with that label. Label union adds the missing label values by creating super-values 6 Sensitive Label Privacy Protection on Social Network Data shared among labels of nodes.

The labels of two or more nodes coalesce their values to a single super-label value, being the union of their values. This approach maintains data integrity, in the sense that the true label of node is included among the values of its label super-value. After such edge insertion and label union operations, if there are nodes in a group still having different neighborhood information, noise nodes with non-sensitive labels are added into the graph so as to render the nodes in group indistinguishable in terms of their neighbors' labels.

We consider the unification of two nodes' neighborhood labels as an example. One node may need a noisy node to be added as its immediate neighbor since it does not have a neighbor with certain label that the other node has; such a label on the other node may not be modifiable, as its is already connected to another sensitive node, which prevents the re-modification on existing modified groups.

Algorithm 1: Global-Similarity-based Indirect Noisy Node Algorithm

Input: graph G (V, E, L, L^s), parameter l;
Result: Modified Graph G'

1 while $V_{left} > 0$ do
2 if $V_{left} \geq l$ then
3 compute pairwise node similarities;
4 group g← v1, v2 with $Max_{similarity}$;
5 Modify neighbors of G;
6 while |g| < l do
7 dissimilarity (V_{left}, G);
8 group G v with $Max_{similarity}$;
9 Modify neighbors of G without actually adding noisy nodes;
10 else if $V_{left} < l$ then
11 for each v∈ V_{left} do
12 similarity (v, G_s);
13 $GMax_{similarity}$←v;
14 Modify neighbors of $GMax_{similarity}$ without actually adding noisy nodes;
15 Add expected noisy nodes;
16 Return G'(V ′,E', L');

In this algorithm, noise node addition operation that is expected to make the nodes inside each group satisfy sensitive-label-diversity are recorded, but not performed right away. Only after all the preliminary grouping operations are performed, the algorithm proceeds to process the expected node addition operation at the final step. Then, if two nodes are expected to have the same labels of neighbors and are within two hops (having common neighbors), only one node is added. In other words, we merge some noisy nodes with the same label, thus resulting in fewer noisy nodes.

V. EXPERIMENTAL EVALUATION

A. Data utility

We compare the data utilities we preserve from the original graphs, in view of measurements on degree distribution, label distribution, degree centrality, clustering coefficient, average path length, graph density, and radius. We show the number of the noisy nodes and edges needed for each approach. The degree distribution of the Facebook graph both before and after modification explained by one algorithm. We can see that the degree distributions of the modified graphs resemble the original ones well, especially when l is small. To sum up, these measurements show that the graph structure properties are preserved to a large extent. The strong resemblance of the label distributions in most cases indicates that the label information, another aspect of graph information, is well maintained. They suggest as well that algorithm GINN does preserve graph properties better than the other two while these three algorithms achieve the same privacy constraint.

B. Information Loss

In view of utility of released data, we aim to keep information loss low. Information loss in this case contains both structure information loss and label information loss. We measure the loss in the following way: for any node v ∈ V, label dissimilarity is defined as:

$$D(l_v, l_{v'}) = 1 - (|l_v \cap l_{v'}| / |l_v \cup l_{v'}|) \quad (2)$$

Where l_v is the set of v's original labels and $l_{v'}$ the set of labels in the modified graph. Thus, for the modified graph including n noisy nodes, and m noisy edges, information loss is defined as

$$IL = \omega 1\, n + \omega 2\, m + (1\ \omega 1\ \omega 2) \sum D(lv, lv') \quad (3)$$

Where $\omega 1, \omega 2$ and 1- $\omega 1$- $\omega 2$ are weights for each part of the information loss. Figure 3 shows the measurements of information loss on the synthetic data set using each algorithm. Algorithm GINN introduces the least information loss.

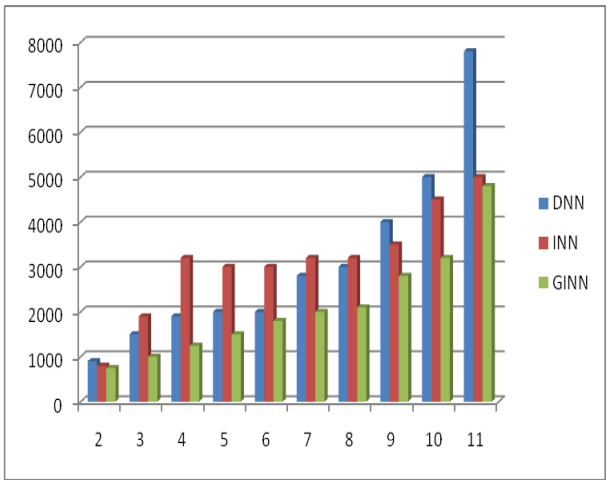

Fig: 3 Information loss.

C. Algorithm scalability

We measure the running time of the methods for a series of synthetic graphs with varying number of nodes in our third dataset. Algorithm DNN is faster than the other two algorithms, showing good scalability at the cost of large noisy nodes added. Algorithm GINN can also be adopted for quite large graphs as follows: We separate the nodes to two different categories, with or without sensitive labels. Such smaller granularity reduces the number of nodes the anonymization method needs to process, and thus improves the overall efficiency.

VI. CONCLUSION

In this paper we have provided privacy for social network data particularly for the sensitive information that

has to be published. We consider graphs with rich label information, which are categorized to be either sensitive or non-sensitive. We assume that adversaries possess prior knowledge about a node's degree and the labels of its neighbors, and can use that to infer the sensitive labels of targets. We suggested a model for attaining privacy while publishing the data, in which node labels are both part of adversaries' background knowledge and sensitive information that has to be protected. In this paper, k-degree-l-diversity model has implemented for privacy preserving social network data publishing. In order to achieve the requirement of k-degree-l-diversity, a noise node adding algorithm to construct a new graph from the original graph with the constraint of introducing fewer distortions to the original graph. Extensive experimental results demonstrate that the noise node adding algorithms can achieve a better result than the previous work using edge editing only. Our experiments on both real and synthetic data sets confirm the effectiveness, efficiency and scalability of our approach in maintaining critical graph properties while providing a comprehensible privacy guarantee.

VII. REFERENCES

[1] L. A. Adamic and N. Glance. The political blogosphere and the 2004 U.S. election: divided they blog. In LinkKDD, 2005.

[2] L. Backstrom, C. Dwork, and J. M. Kleinberg. Wherefore art thou R3579X?: anonymized social networks, hidden patterns, and structural steganography. Com- mun. ACM, 54(12), 2011.

[3] S. Bhagat, G. Cormode, B. Krishnamurthy, and D. S. and. Class-based graph anonymization for social network data. PVLDB, 2(1), 2009.

[4] A. Campan and T. M. Truta. A clustering approach for data and structural anonymity in social networks. In PinKDD, 2008.

[5] J. Cheng, A. W.-C. Fu, and J. Liu. K-isomorphism: privacy-preserving network publication against structural attacks. In SIGMOD, 2010.

[6] G. Cormode, D. Srivastava, T. Yu, and Q. Zhang. Anonymizing bipartite graph
data using safe groupings. PVLDB, 19(1), 2010.

[7] S. Das, O. Egecioglu, and A. E. Abbadi. Anonymizing weighted social network graphs. In ICDE, 2010.

[8] A. G. Francesco Bonchi and T. Tassa. Identity obfuscation in graphs through the information theoretic lens. In ICDE, 2011.

[9] M. Hay, G. Miklau, D. Jensen, D. Towsley, and P. Weis. Resisting structural re-identi_cation in anonymized social networks. PVLDB, 1(1), 2008.

Author's Profile:

Penchalaprasad Kakivai received the B.Tech Degree in Computer Science and Engineering from Narayana Engineering College Gudur, University of JNTUA in 2010. He is currently working towards the Master's Degree in Computer Science, in Annamacharya Institute of Technology and Sciences University of JNTUA. His interest lies in the areas of Data Mining, Networks, Distributed Systems.

S. Prathap Received B.Tech and M.E Degrees in Computer Science and Engineering from priyadharshini college Sullurpet, velmultitech engineering college Chennai. Jntuh, Anna Univesity in 2006&2010 respectively. Currently he is a Assistant Professor in the Department of Computer Science and Engineering at AITS-Tirupati.