Resisting Label-Neighborhood Attacks in Outsourced Social Networks

Yang Wang, Fudong Qiu, Fan Wu, and Guihai Chen
Shanghai Key Laboratory of Scalable Computing and Systems
Department of Computer Science and Engineering
Shanghai Jiao Tong University, China
Email: {cafucwy, fdqiu}@sjtu.edu.cn, {fwu, gchen}@cs.sjtu.edu.cn

***Abstract*—With the popularity of cloud computing, many companies would outsource their social network data to a cloud service provider, where privacy leaks have become a more and more serious problem. However, most of the previous studies have ignored an important fact, i.e., in real social networks, users possess various attributes and have the flexibility to decide which attributes of their profiles are sensitive attributes by themselves. These sensitive attributes of the users should be protected from being revealed when outsourcing a social network to a cloud service provider. In this paper, we consider the problem of resisting privacy attacks with neighborhood information of both network structure and labels of one-hop neighbors as background knowledge. To tackle this problem, we propose a Global Similarity-based Group Anonymization (GSGA) method to generate a anonymized social network while maintaining as much utility as possible. We also extensively evaluate our approach on both real data set and synthetic data sets. Evaluation results show that the social network anonymized by our approach can still be used to answer aggregation queries with high accuracy.**

I. Introduction

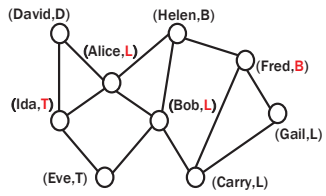
Cloud computing has become a booming computing paradigm in recent years, offering great facilities for storage and computing [1], [2]. It allows companies to migrate their burden (e.g., data maintenance and computing utilities) to a cloud server, which has sufficient resources to maintain very large datasets and provide quick response to customers' requests. Besides, the cloud services are available in a pay as-you-go manner at a low cost. A number of companies have employed cloud computing to publish their large social network data. However, the biggest problem with this approach is privacy disclosure. Therefore, as a cloud service provider, such as Google and Amazon, it is very essential to protect users' privacy from being leaked while keeping these data useful.

Social networks are normally modeled as graphs with nodes and edges, where users are denoted as nodes and social relationships are denoted as edges [3]. Users' privacy could be easily breached by attacks with background knowledge.

This work was supported in part by the State Key Development Program for Basic Research of China (973 project 2014CB340303 and 2012CB316201), in part by China NSF grant 61422208, 61472252, 61272443 and 61133006, in part by CCF-Intel Young Faculty Researcher Program and CCF-Tencent Open Fund, in part by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry, and in part by Program for Changjiang Scholars and Innovative Research Team in University (IRT1158, PCSIRT) China. The opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies or the government.

F. Wu is the corresponding author.

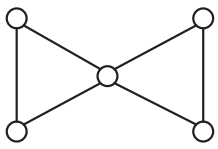
Many approaches have been proposed to protect the privacy of published social networks [14], [15]. These early works mainly concern identity and link disclosures. However, most of them do not consider an important fact that users in real social networks possess attributes, which may be exploited by the attacker to identify a targeted user and should also be well protected.



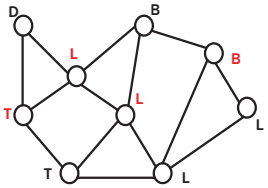
(a) Original social network

Profession	Notation
banker	B
doctor	D
lawyer	L
teacher	T

(b) Table of profession notations



(c) 1-neighborhood graph of Alice



(d) Anonymous social network

Fig. 1. Example of privacy preservation in a social network

In many real-world social networks (e.g., Facebook and LinkedIn), users have plenty of personal information, such as name, gender, age, address and profession. We usually refer to these information as attributes in the users' profile. In the graphs of social networks, attributes are denoted by labels. An individual can select which attributes she wants to conceal in her profile. So labels can be either sensitive or non-sensitive. As a specific example, we consider a synthesized social network of "friends" as shown in Fig. 1(a). Labels attached to the nodes show the professions of the users. For clearance, we use capital letters to represent the professions as listed in Fig. 1(b). Some people do not mind their professions being known by the others, but some do for personal reasons. Therefore the professions are either sensitive (labelled in red) or non-sensitive (labelled in black).

Zhou et al. [4] considered 1-neighborhood attack, where

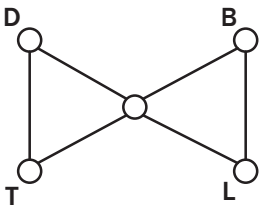


Fig. 2. Alice's label-neighborhood graph

an attacker who has the knowledge of a target's one-hop neighbors and the connections between them, can re-identify the target with high confidence. To resist this attack, they proposed a k-anonymous approach, in which an attacker with the knowledge of any target's 1-neighborhood graph (e.g., Alice's 1-neighborhood graph in Fig. 1(c)) cannot re-identify the target with a confidence higher than $1/k$. As shown in Fig. 1(d), by adding a noise edge between Eve and Carry, the 1-neighborhood graph of every vertex becomes not unique. From this anonymized graph, we can see that, if the attacker with the background knowledge of Alice's 1-neighborhood graph, she cannot identify Alice with probability higher than $1/2$. However, Zhou et al. did not consider labels of users as the background knowledge of the attacker, although the labels are easy to get from users' profiles in the social network.

In this paper, we consider a more comprehensive and practical privacy attack, i.e., label-neighborhood attack, in which an attacker exploits sensitive information based on the background knowledge of 1-neighborhood graph of a target node and the labels of its neighboring nodes. In the 2-anonymized social network in Fig. 1(d), Bob and Carry have the same 1-neighborhood graph as Alice, as shown in Fig. 1(c). Besides this neighborhood structure information, if the attacker also knows that the target has four friends and the jobs of them are B(banker), D(doctor), L(lawyer), T(teacher), respectively, then she can re-identify accurately that which node is Alice and what her profession is. In addition, a k-anonymous social network may still leak sensitive information, when there are not enough diversities in the sensitive attributes. That is to say, if an attacker can link a target to a group of anonymized nodes of which are all associated with the same sensitive attribute, then the attacker can still identify the sensitive attribute of the target. In the above example, if an attacker has the background knowledge of the 1-neighborhood graph of Alice, although Alice, Bob and Carry have the same 1-neighborhood graph in Fig. 1(c), the attacker can recognize exactly the profession of Alice, since the professions of Alice, Bob and Carry are all lawyers.

Given the above problems, we consider the case in which the attackers have the background knowledge of both 1-neighborhood structure and label information (label-neighborhood attack). Our design objective is to protect the users with sensitive attributes from being re-identified and their sensitive information from being disclosed. In this paper, we propose a Global Similarity-based Group Anonymization (GSGA) method to anonymize the original graph of a social network into a graph where any node with a sensitive label is indistinguishable from at least $\ell-1$ other nodes. Our approach consists of two steps, including grouping and anonymizing. We group the nodes in the graph with as similar neighborhood information as possible so that the original graph can be changed as little as possible in the following anonymization step. Meanwhile, we ensure that each group has at least ℓ nodes with different sensitive labels. Then, we propose an effective anonymization algorithm to make suitable modifications to each group to make any node's label-neighborhood graph be isomorphic with at least $\ell-1$ other nodes.

Our contributions are summarized as follows.

- To the best of our knowledge, we are the first to consider the problem of label-neighborhood attack and propose counter-measure, when outsourcing the social network data to a cloud.
- To tackle the problem, we propose a ℓ-diverse approach, namely GSGA, which can prevent users with sensitive attributes from being re-identified as well as their sensitive attributes from being breached.
- We implement and evaluate GSGA on both a real data set and synthetic data sets. Our evaluation results show that the anonymized social network generated by our approach can still be used to answer aggregation queries with high accuracy.

The rest of the paper is organized as follows. In Section II, we introduce our system model and problem definition. We propose our practical solution in Section III. We conduct evaluations on both a real data set and synthetic data sets in Section IV. Then the related work is shown in Section V. Finally, in Section VI, we draw our conclusion and point out possible future work directions.

II. Preliminaries

In this section, we present our system model and give problem definition of privacy preservation in outsourced social network.

A. System Model

We consider a system model that mainly includes four parts, i.e., a cloud service provider, a social network publisher, an attacker, and a set of users. The cloud service provider, such as Google or Amazon, typically has enough resources to hold very large storage spaces and make rapid response to users' requests with its powerful parallel and distributed architecture. The social network publisher, such as Facebook or LinkedIn, usually chooses to outsource their social network data to a cloud platform. The attacker possesses some background knowledge of the target which comprises neighborhood information of both network structure and labels of one-hop neighbors. With this background knowledge, the attacker always wants to analyze the outsourced social network to re-identify the target. The users are considered to be particularly interested in aggregation queries on the social networks [4]. An aggregation query computes the aggregating on selected paths

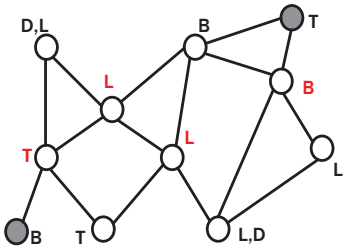


Fig. 3. Privacy-preserved social network

based on some given conditions, e.g., the average shortest distance from a banker to a lawyer in a network.

In our system, we model the social network as a labeled and undirected graph G, in which users are nodes and social connections are edges. To prevent the privacy of users from being disclosed, the social network publisher chooses to anonymize G to G' before outsourcing.

B. Problem Definition

In this paper, we model a social network as $G = (V, V_s, E, L, F)$, where V is the set of all nodes, V_s is the set of nodes with sensitive attributes, E is the set of edges, L is the set of labels, and F is a labeling function, which maps nodes to their labels, $F : V \rightarrow L$. For a graph G, $V(G), V_s(G), E(G), L(G)$, and F_G denote, the set of all nodes , the set of nodes with sensitive attributes, the set of edges, the set of labels and the labeling function in G, respectively. We propose a privacy model, where we assume that an attacker is interested in the identity and sensitive information of a target victim with sensitive attributes. To initiate this attack, the attacker may have background knowledge about the target's label-neighborhood graph, which consists of 1-neighborhood graph of the target and the labels of the target's one-hop neighbors.

In this model, node labels are treated as both a part of an attacker's background knowledge and the sensitive information which we need to protect. Some concepts are clarified by the following definitions:

Definition 1 (1-Neighborhood Graph [4]): For any node u in G, the corresponding 1-neighborhood graph is G_u. $G_u = (V_u, E_u)$, where $V_u = \{v|(u, v) \in E(G) \vee (v = u)\}$, denoting a set of nodes. $E_u = \{(x, y)|(x, y) \in E(G) \wedge \{x, y\} \in V_u\}$, denoting a set of edges.

Definition 2 (Label-Neighborhood Graph): For each node $u \in V(G)$, the related label-neighborhood graph of node u is defined as $G_l(u)$. $G_l(u) = (G_u, NLS_u)$, in which G_u is the 1-neighborhood graph of node u, and NLS_u is a sequence of labels of node u's immediate neighbors.

Definition 3 (ℓ-Diversity [5]): An equivalence class is ℓ-diverse if there are at least ℓ "well-represented" values in it. A table is said to be ℓ-diverse if every equivalence class of the table is ℓ-diverse.

Definition 4 (ℓ-Graphic-Diversity): For each node $u \in V(G)$ that attaches with a sensitive label, there must be at least $\ell - 1$ other nodes with the same label-neighborhood graph, but possesses different sensitive labels.

The privacy issue in this paper is mainly from the disclosure of sensitive labels. To protect the sensitive attributes of users satisfactorily, one might recommend that such labels should be simply removed. However, such a way would result in a partial view of the real social network, and would hide some valuable statistical information which does not breach users' privacy. A more sophisticated approach is to release these sensitive attributes about the users, while guaranteeing that the identities and sensitive attributes of these users cannot be revealed. In this paper, we ensure that the identity and sensitive attributes of any individual with sensitive attributes cannot be identified correctly in the anonymized social network with a probability higher than $1/\ell$, where ℓ is a user-specified parameter carrying the same meaning in the ℓ-diversity model [5]. In Fig. 1(a), node Alice, Bob, Ida and Fred have sensitive labels, by adding noise nodes and noise edges, and merging some nodes' labels. The graph in Fig. 3 satisfies 2-graphic-diversity. That is because, in this graph, node Ida and node Bob have four neighbors with label $B, \{D, L\}, L, T$ respectively, and the same neighborhood structure, so they are indistinguishable. Likewise, node Alice and node Fred are indistinguishable, as they also have the same label-neighborhood graph.

III. Design of GSGA

In this section, we propose a Global Similarity-based Group Anonymization (GSGA) approach to anonymize an outsourced social network. The approach mainly consists of two key steps. The first step is to make suitable grouping for nodes. We want to group nodes with as similar label-neighborhood graph as possible so that we can change the graph of original social network as little as possible. The second step is to make appropriate modifications to each group to satisfy the ℓ-graphic-diversity requirement.

A. Node Grouping

A good grouping contributes significantly to reduce the costs of modification on the graph. We group nodes by using the metric: neighborhood label sequence similarity $(NLSS)$ [7]. For two nodes v_1 with neighborhood label sequence (NLS_{v_1}), and v_2 with neighborhood label sequence (NLS_{v_2}), their neighborhood label sequence similarity can be calculated as follows:

$$NLSS(v_1, v_2) = \frac{|NLS_{v_1} \cap NLS_{v_2}|}{|NLS_{v_1} \cup NLS_{v_2}|}, \quad (1)$$

The larger the value is, the larger similarity between the two nodes' neighborhood label sequences is. Some other metrics, e.g., neighborhood structure, clustering coefficient and nodes degree, can also be used for grouping. Since our anonymized social network is mainly used to answer aggregation queries, we only use the metric mentioned above. We verify experimentally that it is very effective to modify graph by utilizing the above metric to divide nodes into groups.

Our approach is illustrated in Algorithm 1. The algorithm generates the groups, such that each group's size is at least ℓ. Here, we process nodes in the degree descending order, the nodes with sensitive labels that have not yet been grouped are

Algorithm 1: GROUPING(G)

Input: A social network G, the privacy parameter ℓ;
Output: A group set C;

```
1  Sort(V);
2  V_s = the nodes with sensitive label in V;
3  while V_s ≠ ∅ do
4      u_s = the first node in V_s;
5      V = V − {u_f};
6      group g = new group {u_s};
7      while |g| < ℓ do
8          Candidates = ∅;
9          for Each node u ∈ V do
10             if u.label does not be included in g then
11                 candidates = candidates ∪ {u};
12         if |candidates| > 0 then
13             for Each node u ∈ candidates do
14                 u_max = arg max_{u∈V} NLSS(u, u_s);
15             g = g ∪ {u_max};
16             V = V − {u_max};
17         else
18             break;
19     C = C ∪ {g};
20     if |g| < ℓ then
21         C = C − {g};
22         for Each node u ∈ g do
23             g' = the group in C with the maximum label
               similarity with u;
24             g' = g' ∪ {u};
```

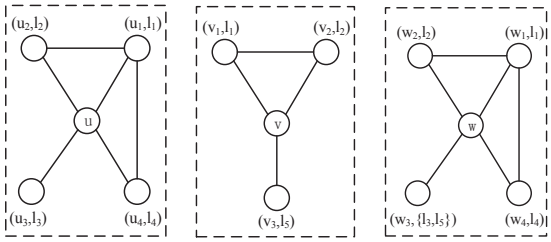
first taken into account. Iteratively, we choose the first node u_f in V_s and create a new group g for this node (line 4-6). Then in line 7-19, the nodes having the maximum neighborhood label sequence similarities with node u_f in the group are clustered into the group until the group has ℓ nodes with different labels. If the size of group g cannot reach ℓ in line 20-24, we first remove this group, then for each node in this group, we insert it into an existing group which has the maximum (estimated) neighborhood label sequence similarity with the node. Finally, when the algorithm terminates, we can get the node group set C where each group size is at least ℓ.

B. Anonymization

After grouping, we need to design an anonymization algorithm to ensure that the nodes in each group are indistinguishable in terms of their label-neighborhood graph. Before anonymization, we first state an important property in the large social network. The property has been acknowledged in different kinds of social networks including friends networks, collaboration networks, and email networks, and can help us to design proper anonymization approach.

In the real social network, the degrees of nodes usually follow *the power law distribution* [6], which indicates that most of the node degrees are relatively low, and only a small number of nodes in practice have high degrees.

Since the degrees of nodes in a large social network follow the above property, we first process those nodes with high degrees, which can maintain the information loss low and keep high quality in the anonymization. Besides, since it is relatively easy to anonymize those low degree nodes, we can utilize these low degree nodes to anonymize high degree nodes. So our anonymization algorithm processes nodes in the degree descending order.



(a) $G_l(u)$ (b) $G_l(v)$ (c) $G_l(w)$

Fig. 4. Anonymizing the label-neighborhood graphs of two nodes

1) Anonymization cost: Information loss in our anonymization model of social network contains both structure information loss and label information loss. To modify graph with as small information loss as possible, we devise three modification operations: label generalization, edge insertion and node insertion.

Label generalization is to make the nodes within each group have the same neighborhood label sequence. In order to keep the nodes' labels distribution of each group unchanged in the anonymized graph, we do not generalize all the labels to be the same value. Instead, we add the missing labels by creating a super-label which is composed of several label values of nodes in the group. In this sense, a super-label contains the true label value of a node. Edge insertion is to keep node's neighborhood structure similar. After such edge insertion and label generalization operations, if we cannot still make nodes indistinguishable in terms of label-neighborhood graph, nodes with non-sensitive label need to be inserted into the graph in order to make the nodes' label-neighborhood graphs in each group isomorphic.

Each of the above three operations can lead to some information loss, we measure the loss in the following way: for any node $u \in V$, label generalization cost is :

$$LGC(l_u, l'_u) = 1 - \frac{|l_u \cap l'_u|}{|l_u \cup l'_u|}, \tag{2}$$

where l_u is the set of u's labels in the original graph and l'_u is the set of labels in the anonymized graph. The information loss due to adding edge and node can be measured by the total number of edges added and nodes added, separately.

We consider a social network G are anonymized to G'. For two nodes $u, v \in V(G)$, let $T = G_l(u) \cup G_l(v)$ and

Algorithm 2: ANONYMIZATION ALGORITHM

Input: A social network G, the parameters α, β and γ;
Output: An anonymized social network G';

```
1  G' = G;
2  C = grouping(G);
3  CandidateSet = ∅;
4  Sort C in descending order of the number of nodes;
5  mark each node in V(G') and each group in C as
   "unanonymized";
6  while |C| > 0 do
7      g_f = the first group in C and remove it from C;
8      u_f = the first node in g_f and remove it from g_f;
9      for Each node u_i ∈ g_f do
10         Using Eq. (3) to calculate cost(u_f, u_i);
11         u_i.cost = cost(u_f, u_i);
12     Sort nodes of group g_f in ascending order of the cost
       value of each node;
13     for Each node u_i ∈ g_f do
14         Using the method in Section III-B.2 to
           anonymize G_l(u_f) and G_l(u_i);
15         mark node u_f and node u_i as "anonymize";
16     mark group g_f as "anonymize";
17     CandidateSet = CandidateSet ∪ {g_f};
18     for Each group g_i ∈ CandidateSet do
19         if g_i is unanonymized then
20             remove it from CandidateSet and insert it
               into C;
```

$T' = G'_l(u) \cup G'_l(v)$. The anonymization cost is defined as

$$\begin{aligned} cost(u,v) &= \alpha \cdot \sum\nolimits_{u \in T} LGC(l_u, l'_u) \\ &+ \beta \cdot (|E(T')| - |E(T)|) \\ &+ \gamma \cdot (|V(T')| - |V(T)|), \end{aligned} \tag{3}$$

where α, β and γ are weights associated with each part of the information loss.

The similarity between node u's label-neighborhood graph and node v's label-neighborhood graph is measured with their anonymization cost. The smaller the anonymization cost is, the more similar the two label-neighborhood graphs are.

2) Anonymizing two label-neighborhood graphs: Consider two nodes u, $v \in V(G)$, $G_l(u)$ and $G_l(v)$ are the label neighborhood graphs of u and v, respectively. A greedy approach is proposed to anonymize $G_l(u)$ and $G_l(v)$.

This approach conducts a label-first and degree-later matching. The node matching procedure is processed in the descending order of node degrees in $G_l(u)$. First, the nodes in $G_l(u)$ are matched with the nodes in $G_l(v)$ such that they have the same label. If multiple nodes in $G_l(v)$ are found, we select a node whose degree is the closest to the degree of that unmatched node in $G_l(u)$. Then, we consider the remaining unmatched nodes in $G_l(u)$ that have the same degree as the nodes in $G_l(v)$. If there are no nodes in $G_l(u)$ with the same degree as nodes in $G_l(v)$, again, we relax the matching condition and pick a node in $G_l(v)$ whose degree is the closest to the degree of the unmatched node in $G_l(u)$. Finally, if there are still some unmatched nodes in $G_l(u)$, some unanonymized nodes in $V(G)$ which have the same labels as these unmatched nodes are added into $G_l(v)$, and are matched in pairs. If we cannot find such nodes in $V(G)$, we create some new nodes which attach the same labels as these unmatched nodes to add them into $G_l(v)$. When all the nodes are matched, we insert some edges into $G_l(u)$ and $G_l(v)$ to make them isomorphic. The anonymization cost of two nodes can be calculated according to this matching procedure.

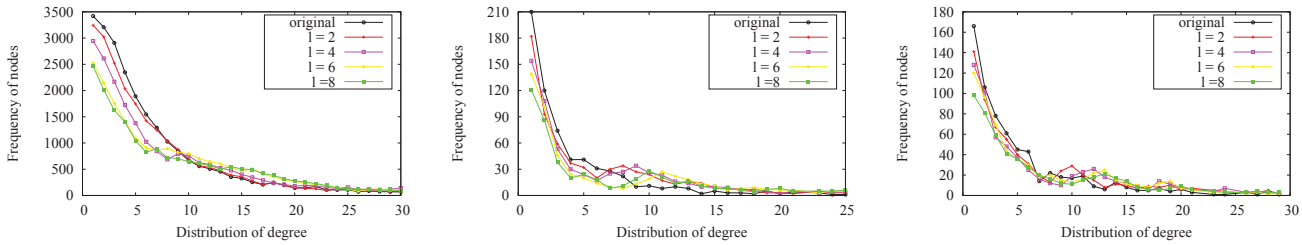
For example, the label-neighborhood graphs $G_l(u)$ and $G_l(v)$ of two node u and v are shown in Fig. 4. Each neighbor node of u and v is denoted in the form of $(id, label)$. According to the above matching procedure, we can find that, node u_1, u_2, u_3 match with node v_1, v_2, v_3, respectively. However, node u_4 cannot find any node matching in $G_l(v)$, so we add a node which has the same label as node u_4 into $G_l(v)$. Finally, we insert some edges into $G_l(u)$ and $G_l(v)$ so that they are anonymized to the same, namely, $G_l(w)$. In this example, the anonymization cost of node u and v is $\alpha \cdot 1 + \beta \cdot 2 + \gamma \cdot 1$.

3) Anonymizing the social network: Suppose that we get the group set C after grouping, where each group size is at least ℓ. We propose a greedy algorithm to anonymize a social network as shown in Alg. 2.

First, we sort group set C in descending order based on the number of nodes and mark all the nodes and groups as "unanonymized" (line 4-5). Each time, we pick the first group in C as the processing group and the first node in the first group as the seed node. Then in line 9-12, for each node u in the processing group, we use Eq. (3) to calculate the anonymization cost of the seed node and node u, then we sort nodes in ascending order of the anonymization cost values.

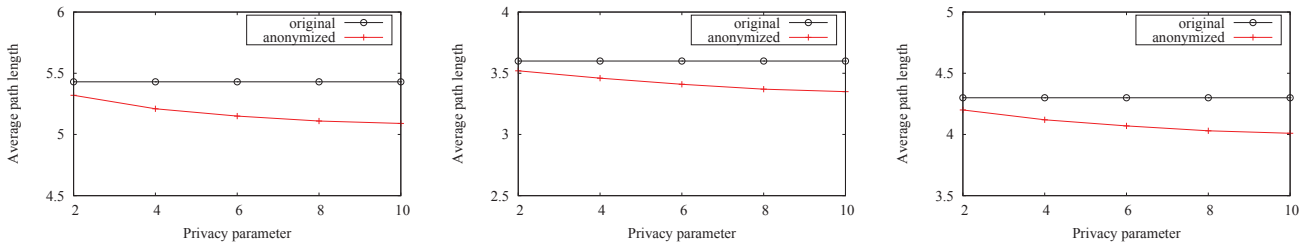
In line 13-17, the label-neighborhood graphs of seed node and each node in the processing group are anonymized to the same in turn, then we mark them as "anonymized". In order to maintain the ℓ-graphic-diversity for a group of nodes, any change to the label-neighborhood graph of seed node will be also put into use in the previous anonymized nodes. After anonymizing all nodes in the processing group to the same, we mark the group as "anonymized" and insert it into the $CandidateSet$. During anonymizing, it may cause some other nodes that have been marked as "anonymized" in another group (e.g., label generalization and edge insertion between an unanonymized node and an anonymized node) to be altered. Once those nodes are changed, they and the groups that those nodes belong to are marked as "unanonymized". Finally, in line 18-20, We remove those groups which are marked as "unanonymized" from $CandidateSet$ and insert into the group set C again. The anonymization algorithm stops when all the nodes in the soical network graph are marked as "anonymized".

In this algorithm, when we anonymize two nodes, label generalization and edge insertion are better than node insertion, as they can give rise to less variation to the overall structure of the graph. Since we only anonymize the nodes



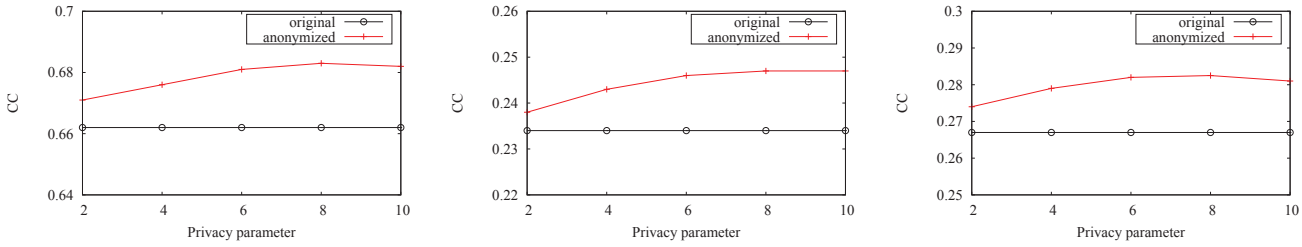
(a) ca-CondMat (b) Systhetic dataset (average node degree=4) (c) Systhetic dataset (average node degree=6)

Fig. 5. Distribution of node degrees



(a) ca-CondMat (b) Systhetic dataset (average node degree=4) (c) Systhetic dataset (average node degree=6)

Fig. 6. Average shortest path length



(a) ca-CondMat (b) Systhetic dataset (average node degree=4) (c) Systhetic dataset (average node degree=6)

Fig. 7. Clustering co-efficient

with sensitive labels, for the purpose of anonymization, many nodes with non-sensitive labels can be used to link to the anonymized nodes. So the modification operation about node insertion is rarely used. Besides, due to the important property of real social networks (node degree in power law distribution), our algorithm can stop very rapidly in practice, and the total anonymization cost is relatively small.

IV. Evaluation

In this section, we evaluate our anonymization approach on both a real data set and two synthetic data sets. All the evaluations run on a desktop with Intel Core(TM) i3-2330M 2.20GHz, 4G RAM, and Windows 7 Ultimate operating system.

A. Data sets

1) Real Data set: We use a real data set (ca-CondMat) to validate the performance of our anonymization algorithm. This real data set presents a Arxiv COND-MAT (Condense Matter Physics) collaboration network [10]. It covers scientific collaborations between authors, submitted papers to Condense Matter category in the period from January 1993 to April 2003, and contains 23133 nodes and 186936 edges. The average degree of nodes is 8.08. For two author i and j, a undirected edge from i to j is created in the graph if author i co-authored a paper with author j. If the paper is co-authored by k authors, a completely connected (sub)graph on k nodes will be generated.

Due to the lack of node labels in the ca-CondMat, we use a random number generator to generate what we need. First, we assign a uniformly distributed random number in the range [0,1000] to each node as its label. Then, we also use random number generator to set half of the total number of nodes with sensitive attribute, and the remainder nodes are with non-sensitive attribute.

2) Synthetic Data sets: We use Pajek [11] to generate some random graphs with scale-free property for large network analysis. The degree of a scale-free network follows the power law distribution, at least asymptotically. In our experiments, the default number of nodes is 1000, and we use two kinds of graphs with different average node degree in synthetic data

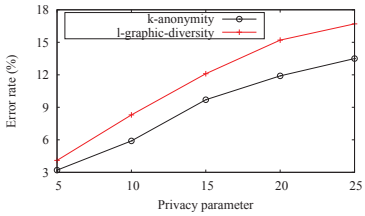
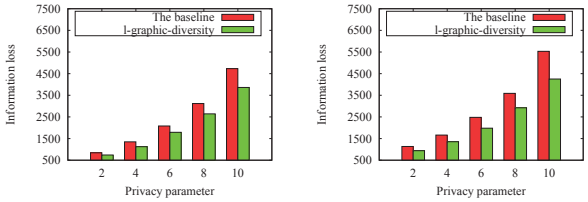


Fig. 8. Aggregate query result on the ca-CondMat dataset



(a) Average node degree = 4 (b) Average node degree = 6

Fig. 9. Information loss on various synthetic datasets

sets, namely 4 and 6. In order to assign a label to each node and set whether it is a sensitive label or not, we use the same method as the above in the real data set.

B. Data Utilities

The data utilities are mainly based on the preservation of some graph properties. In our experiment, we adopt several measurements to measure the utilities of the anonymized graph. The first measurement is degree distribution, which is the probability distribution of degrees of all nodes over the whole network. The second measurement is average path length. It is defined as the average number of steps along the shortest paths for all possible pairs of network nodes. The last measurement is clustering coefficient, also known as transitivity. It reflects the degree to which nodes in a graph tend to cluster together. We test these measurements on real data set and synthetic data sets, respectively. Besides, we test on the real data set whether the anonymized graph can be used to answer aggregation queries.

Fig. 5-7 show some graph properties and experimental results for the two data sets. In Fig. 5, we compare the distribution of node degrees in original graph with that in the anonymized graph with $l = 2, 4, 6, 8$. We observe that, the number of nodes with low degree in anonymized graph is smaller than the number in the original graph, especially when the degree is 1 and 2. The reason is that our anonymization algorithm processes nodes in the degree descending order, and many low degree nodes are used to anonymize those nodes with high degree. From the overall view, the degree distributions between the anonymized graph and the original graph are very similar, especially when l is small. Fig. 6 shows the results of average shortest path length in the three graphs. As the edge insertion is conducted during anonymization, the average shortest path length of anonymized graph decreases slightly. However, the value in the anonymized graph is still close to the original value. We also plot the clustering co-efficient (CC) of the anonymized graph and original graph in Fig. 7. The clustering co-efficient in the anonymized graph is tending towards stability when $l = 8, 10$. This is because, some new nodes are created to anonymize the graph as l increases. Besides, We can see that the clustering co-efficient distributions between the anonymized graph and the original graph are quite similar. Even then $l = 8$, the difference is only 0.021 in the ca-CondMat data set.

To further evaluate the utility of the anonymized graph, we conduct the aggregation queries on the ca-ConMat data set. For any two different labels l_1 and l_2, we calculate the average shortest path length from the node with label l_1 to the node with label l_2. If d and d' are denoted as the average shortest path length in the original graph and in the anonymized graph, separately, the error rate is $(d - d')/d$. Since some labels are generalized to super-label in the anonymized graph, we adopt a probabilistic approach to randomly sample a graph which is consistent with the anonymized graph [19]. That is, for each node with super-label, we choose a random label from this super-label to assign the node as its label. We conduct this process several times to generate 10 sample graphs and perform the queries over these resulting graphs to get a average value. We randomly choose 20 different label pairs from the data set, then calculate their average error rate. The related results are shown in Fig. 8. We compare our approach with k-anonymity method [4]. The result of k-anonymity is slightly better than our approach. However, it cannot protect nodes' sensitive attributes. From Fig. 8, we can see that the error rate is relatively small even when l is up to 25.

C. Information Loss

To measure the information loss during anonymization, we first need to set the parameters α, β and γ. We use the same method as [4] to test the impact of the three parameters on the anonymization cost. In this method, we set β to 10 as the base, and vary the values of α and γ to measure the label generalization cost and the number of nodes added on the synthetic data sets. Due to space limitation, we omit some experimental results, such as the tradeoff between label generalization and nodes added. We see that, when $\alpha = 65$, $\beta = 10$ and $\alpha = 25$, the total anonymization cost is minimum.

We compare information loss of our approach and the following baseline algorithm. In this algorithm, when anonymizing two nodes' label-neighborhood graphs, we use a degree-first and label-later method to try to match nodes in the two label-neighborhood graphs. The other parts of this algorithm are the same with our algorithm in this paper.

We test anonymization costs on the two synthetic data sets, and the parameters α, β and γ in two algorithms are set to 65, 10 and 25, respectively. The related results are shown in the Fig. 9. We can observe that, our approach surpasses the baseline algorithm. When l increases, the information loss also increases. Besides, when the average node degree increases, the information loss increases as well. This is because, the nodes' label-neighborhood graphs are more complicated in the

dense graph and we need more edges and nodes to anonymize different label-neighborhood graphs.

V. Related Works

Privacy protection for social network data was first presented in [9], where they showed that naive anonymization is insufficient to protect users from privacy leaking, as the structure of the published social network might reveal the identity of the individuals, and discussed both active and passive attacks using small subgraph. Hay et al. [12] emphasized this problem and quantified the risk of re-identification, then they proposed that we need to protect privacy against this subgraph background knowledge. However, they did not provide a solution to counter these attacks. Being aware of this problem, several works in [13]–[18] proposed methods that could be used to anonymize the social network graph while providing certain privacy guarantee.

Most of the above works in privacy preserving publishing of social network aim at the issue of node re-identification, which means that the adversary is not able to link any individual to a node with high confidence in the published social network. Zhou and Pei [4], [20] and Yuan et al. [21] first considered that social networks were modeled as labeled graphs, which is similar to what we consider in this paper. To resist re-identification attacks by attackers with one-hop neighborhood structural background knowledge, Zhou and Pei [4] proposed a k-anonymity method that each node must have at least $k-1$ others with the same neighborhood structure, and they focused on the case of labels drawn from a hierarchy. However, as with the situation of microdata, a social network graph that satisfies a k-anonymity privacy requirement may still leak sensitive information. This problem had been identified in [5]. In [20], to protect the textual attributes of users as well, the idea of ℓ-diversity was introduced by them to offer stronger privacy guarantee. However, Zhou and Pei did not have a systematic introduction to ℓ-diversity method. Besides, they have different problem background with us in this paper. Yuan et al. [21] considered that users have different privacy demands, and classified privacy requirements into three levels. To every level privacy demand, they designed corresponding methods to anonymize the social network graph. Nevertheless, they did not consider neighborhood information of both network structure and labels of neighbors as background knowledge possessed by the attackers. Once attackers have label information of nodes, their methods cannot achieve the same privacy guarantee as ours.

Our work is mainly to study and solve privacy issues when outsourcing a social network to a cloud. The area is still in its initial stage. To our best knowledge, the work in [22], [23] were the first to consider this problem in this area. However, they only devoted to the issue of node re-identification. In our work, we also protect the sensitive attributes of users from being disclosed and identify a more comprehensive and practical attack model. Our method can offer stronger privacy guarantees.

VI. Conclusion and Future Work

In this paper, we have studied the issues on the protection of the identities and sensitive attributes of users when publishing an outsourced social network in a cloud. We have considered a more comprehensive and reasonable attack, namely label-neighborhood attack, where attackers possess the background knowledge about a node's neighborhood structure and the labels of its neighbors, and can utilize this background knowledge to infer the identity and sensitive labels of targets. To resist this attack, we have proposed GSGA to prevent users sensitive information from being leaked. The approach offers stronger privacy guarantees than existing work. Our evaluation results on both real data set and synthetic datasets have indicated that the social network anonymized by our GSGA method can still be used to answer aggregation queries with satisfying accuracy. For future work, we will consider how to protect against d-label-neighborhood ($d > 1$) attacks. Moreover, we will try to introduce stronger privacy methods to protect the privacy in outsourced social network. e.g., t-closeness and differential privacy.

References

[1] D. J. Abadi. Data management in the cloud: Limitations and opportunities. *IEEE Data Eng. Bull. (DEBU)*, 32(1):3-12, 2009.

[2] H. Hacigms, B. R. Iyer, and S.Mehrotra. Providing database as a service. In *ICDE*, 2002.

[3] B. Zhou, J. Pei, and W. Luk. A brief survey on anonymization techniques for privacy preserving publishing of social network data. In *ACM SIGKDD Explorations Newsletter*, 2008.

[4] B. Zhou and J. Pei. Preserving privacy in social networks against neighborhood attacks. In *ICDE*, 2008.

[5] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. L-diversity: privacy beyond k-anonymity. In *ICDE*, 2006.

[6] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. In *SIGCOMM*, 1999.

[7] Y. Song, P. Karras, Q. Xiao, and S. Bressan. Sensitive label privacy protection on social network data. In *SSDBM*, 2012.

[8] L. Sweeney. k-anonymity: a model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557-570, 2002.

[9] L. Backstrom, C. Dwork, and J. Kleinberg, Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *WWW*, 2007.

[10] COND-MAT, http://snap.stanford.edu/data/ca-CondMat.html.

[11] Program for Analysis and Visualization of Large Networks(pajek), http://pajek.imfm.si/doku.php.

[12] M. Hay, G. Miklau, D. Jensen, D. Towsley, and P. Weis. Resisting structural reidentification in anonymized social networks. In *VLDB*, 2008

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

[14] K. Liu and E. Terzi. Towards identity anonymization on graphs. In *SIGMOD*, 2008.

[15] C. Tai, P. Yu, D. Yang, and M. Chen, Privacy-preserving social network publication against friendship attacks. In *SIGKDD*, 2011.

[16] L. Zou, L. Chen, and M. TamerÖzsu. K-automorphism: a general framework for privacy preserving network publication. In *VLDB*, 2009.

[17] F. Bonchi, A. Gionis, and T. Tassa. Identity obfuscation in graphs through the information theoretic lens. In *ICDE* 2011.

[18] G. Cormode, D. Srivastava, T. Yu, and Q. Zhang. Anonymizing bipartite graph data using safe groupings. In *VLDB*, 2008.

[19] G. Cormode, D. Srivastava, S. Bhagat, and B. Krishnamurthy. Class-based graph anonymization for social network data. In *VLDB*, 2009.

[20] B. Zhou and J. Pei. The k-anonymity and ℓ-diversity approaches for privacy preservation in social networks against neighborhood attacks. *Knowledge and Information Systems*, 28(1):47-77, 2010.

[21] M. Yuan, L. Chen, and P. S. Yu. Personalized privacy protection in social networks. In *VLDB*, 2010.

[22] J. Gao, J. Yu, R. Jin, J. Zhou, T. Wang, and D. Yang. Neighborhood privacy protected shortest distance computing in cloud. In *SIGMOD*, 2011.

[23] G. Wang, Q. Liu, F. Li, S. Yang, and J. Wu. Outsourcing privacy-preserving social networks to a cloud. In *INFOCOM*, 2013