Preserving Privacy in Social Networks Against Label Pair Attacks

Chenyang Liu, Dan Yin(✉), Hao Li, Wei Wang, and Wu Yang

Information Security Research Center,
Harbin Engineering University, Harbin 150001, China
{chenyliu, yindan, lhao, w_wei, yangwu}@hrbeu.edu.cn

Abstract. With the popularity of social networks, publishing social network data is necessary for research purposes, which causes privacy leakage undoubtedly. Therefore, many methods are proposed to deal with different attack models. This paper focuses on a novel privacy attack model and refers it as a label pair attack. In the label pair attacks, the adversary can re-identify a pair of friends by using the labels of two vertices connected by an edge. We present a new anonymity concept, called Label Pair k^2-anonymity which ensures that there exists at least $k - 1$ other vertices such that each of the $k - 1$ vertices also has an incident edge of the same label pair and reduces the probability of a vertex being re-identified to less than $1/k$. The experimental results demonstrate that the approach can preserve the privacy and utility of social networks effectively.

Keywords: Privacy preserving · Social network · Label pair

1 Introduction

More and more social network datasets are published for different purposes, such as research purposes with the advance on mobile and Internet technology. Specially, the mobile social network is popular among people. It provides a platform for sharing interests, hobbies, status and activity information. So the publication of social network datasets may lead the privacy leakage easily. This problem has raised people's attention, many works [1, 5–7, 20, 22, 23] have proposed various protection means to protect individual privacy from attack. The social networks are modeled as a graph in which each vertex represents a user, each edge represents the social relationship and the label indicates the feature of one user.

There are a variety of attacks nowadays, such as friendship attacks, mutual friend attacks, neighborhood attacks and structural attacks. Some works solve the social structure problem only, and some works solve the problem of social networks with users' labels. We consider the two aspects of the structure and labels. Whereafter, we propose a new attack named label pair attacks. The attacks frequently happen in mobile social networks. The adversary can use the vertex labels of two individuals and friendship to identify users. And the labels and friendship of users are easily obtained, the adversary can easily launch the attack. However, the methods referred in papers [2, 11, 13, 14, 18] can protect privacy from common attacks, while they cannot protect

L. Ma et al. (Eds.): WASA 2017, LNCS 10251, pp. 381–392, 2017.
DOI: 10.1007/978-3-319-60033-8_34

privacy from the label pair attacks. Thus it can be seen that the problem should be solved as soon as possible.

In this paper, we introduce a new relationship attack model based on the vertex label pair of an edge. An adversary can acquire the label of an individual from the social network website or application easily, such as Facebook, Twitter. Furthermore, the adversary can also know whether two individuals have a friendship relation. The label pairs which will be mentioned later are made of the labels of two individuals who are friends. So the adversary can use the label pairs to issue a label attack from the published social networks on the purpose of recognizing victims' identity. As a concrete example, Fig. 1(a) is an original social network, every vertex represents a user, such as Mary, Bob, Ed. Meanwhile, we can obtain the profession label of each user, such as Doctor, Teacher. Then we remove all users' names and reserve the profession labels as shown in Fig. 1(b). Obviously, an adversary cannot re-identify anyone from the social network with anonymous vertices with the only label information. But if the adversary knows Mary's label is Doctor, Bob's label is Teacher and they have a friendship relation, he can easily identify Mary and Bob through the label pair (Doctor, Teacher) in Fig. 1(b). Only if an adversary grasps the background knowledge about friendship relation and the label information, he can launch the attack to identify individuals and obtain various privacy information.

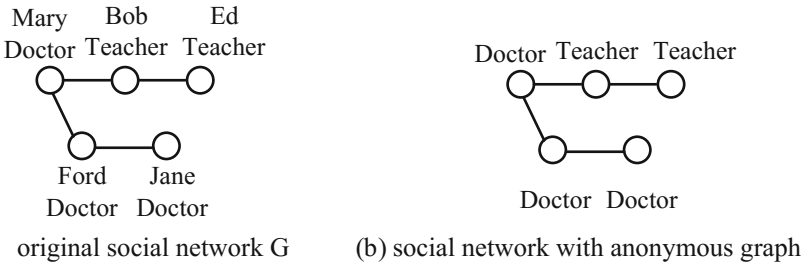


Fig. 1. An example of the label pair attacks.

To avoid the label pair attacks, a new type of privacy-preserving method called LP k^2-anonymity is introduced in this paper. For every vertex v with an edge of label pair (l_1, l_2), there will be at least $k-1$ other vertices having an edge of the same label pair. It can be guaranteed that the probability of a vertex being identified is not greater than $1/k$. We propose algorithms to achieve LP k^2-anonymity for the graphs of original social networks. Our approach mainly includes two steps. First, we adopt a method named LGA (Label Generalization Anonymization) to group the vertices and generalize the labels of vertices. Then we anonymize the graph by LGAN (Label Group ANonymization). The algorithm can effectively protect the individual privacy from the label pair attacks, meanwhile preserve the vertex set. Above all, the algorithm is designed to preserve as much utility as possible.

Contributions. Our contributions are summarized as follows:

1. This paper is the first to propose the new type attack model named label pair attacks. And we take measures to tackle the problem of the label pair attacks.
2. To deal with the problem, we introduce Label Pair k^2-anonymity concept, namely LP k^2-anonymity, which can prevent users with labels from being re-identified when the adversary launches the label attack.
3. Two algorithms are devised to achieve the purpose to anonymize. The first algorithm is to group the vertices and generalize the labels of vertices named LGA (Label Generalization Anonymization). Another algorithm named LGAN (Label Group ANonymization) is to anonymize the social networks by edge addition and edge deletion. We do not add noise vertices or delete vertices to preserve the vertex set and adopt the specific order which is illustrated in Sect. 4 to protect dataset utility.
4. The empirical results on the real datasets show that our algorithms perform well in anonymizing the social networks.

The rest of paper is organized as follows. We introduce the related work about the problem of anonymizing social networks in Sect. 2. We define the problem and propose the practical solution in Sects. 3 and 4. Finally, we conduct the experiments on real data sets and conclude in Sects. 5 and 6.

2 Related Work

Privacy preservation in publishing social networks is a new challenge that has drawn more and more people's attention. Recently, some works [15] propose to encrypt for the data to protect privacy and other works propose to achieve k-anonymity based on various adversary knowledge. Many approaches [5, 10] have been proposed to guarantee privacy. Liu and Terzi [8] propose the k-degree anonymization, for any node v, there exists at least $k - 1$ other nodes in the graph having the same degree as node v. Sun et al. [11] propose a new type anonymity concept, called k-NMF anonymity against mutual friend attacks. Zou et al. [16] propose the k-automorphism model, which converts the original network into a k-automorphic network. Tai et al. [2] present a friendship attack, in which the adversary uses the degree of each vertex and friendship relation to identify users. Zhang et al. [21] combine k-anonymity and randomization together to protect data privacy.

What is said above is not referred the data with labels. There are labels of vertices and labels on the edges. Liu et al. [9] treat weights on the edges as sensitive labels and propose a method to preserve shortest paths between most pairs of vertices in the graph. And some studies usually generalize labels [3, 4, 12] to protect privacy. Generalization involves replacing (or recording) a value with a less specific but semantically consistent value. Zhou and Pei [18] adopt this way in social networks. They propose a practical solution to battle neighborhood attack, the solution considers modeling social networks as labeled graphs and also can be used to answer aggregate network queries with high accuracy. Yuan et al. [13] introduce a framework which provides privacy protection

based on the users' requests. It combines the label generalization protection and the structure protection techniques to satisfy three levels' requests. Song et al. propose a privacy protection scheme that only prevents the disclosure of identity of users but also the disclosure of sensitive labels. Yuan et al. [14] define a *k*-degree-*l*-diversity anonymity model that consider the protection of structural information as well as sensitive labels of individuals and further propose a novel anonymization method based on adding noise vertices.

3 Preliminaries and Problem Definition

In this paper, we model a social network as an undirected graph $G = (V, E, L)$ where V is a set of nodes which represents the individuals, $E \subseteq V \times V$ is a the set of edges representing the relationship of users, and L is a set of labels. In this work, we assume that the adversary uses friendship relations and labels of users as background knowledge to reveal the identities of users. First, we should form a generalization tree (GTree) using the label set L. For example, if the locations of users are used as labels of vertices in a social network, L contains not only the specific locations such as Beijing, Washington, New York, California, Berlin, London, but also general categories like China, America, Germany, England. We assume that there exists a symbol $* \in L$ which is the most general category generalizing all labels. For two labels $m, n \in L$, if m is more general than n, we write $m \prec n$. For example, America $\prec$ New York. And when we form the generalization tree (GTree), we had better form it by the number of leafs in descending order. We put nodes which have descendants in front of the nodes having fewer on the purpose of reducing cost and protect the utility of data. These concepts are clarified by the following definitions:

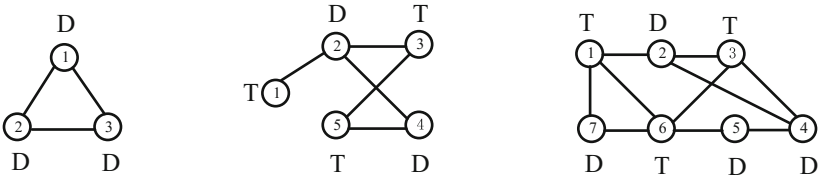
Definition 1 Label Pair Attack. Given a social network $G = (V, E, L)$ and the anonymized network $G' = (V', E', L')$ for publishing. For a vertex $v \in V$ and all edges connecting with it, the adversary can get the label pair (m, n) corresponding one edge. The adversary can take advantage of the label pair to identify victims.

An adversary re-identifies the v with high confidence if the number of candidate vertices is too small. Hence, we set a threshold k to make sure that the number of candidate vertices is no less than k for each vertex $v \in V$. We define LP k^2-anonymity as follows.

Definition 2 LP k^2-Anonymity. If a graph $G' = (V', E', L')$ is LP k^2 anonymous, for each vertex with an edge of label pair (m, n) in G', there exists at least $k - 1$ other vertices having an edge of the same label pair.

Consider the graphs in Fig. 2 as an example. Each vertex in the graphs represents a user, the edge between two vertices represents the fact that the two users are friends. And the labels annotated to the vertices show the profession of the user. For convenience, we make letter T represent the profession of teacher and D represent the profession of doctor. The Fig. 2(a) is a simple example. There are three vertices and each vertex has the label pair (D, D). Therefore, the graph is $LP\,3^2$ anonymous. The Fig. 2(b) is $LP2^2$ anonymous. Because vertices {1, 4} have the label pairs (D, T) and

(D, D), vertices {2, 3, 5} have the label (T, D) and vertices {2, 3} have the label pair (T, T). Similarly, in the Fig. 2(c), vertices {1, 3, 5} have the label pairs (T, D) and (T, T), vertices {2, 4, 6} own the label pair (D, T) and vertices {2, 6, 7} own the label pair (D, D). Hence, it is $LP\,3^2$ anonymous.



(a) $LP\,3^2$ - *anonymity* (b) $LP\,2^2$ - *anonymity* (c) $LP\,3^2$ - *anonymity*

Fig. 2. Examples of LPk^2 - *anonymity* graphs

Definition 3 Generalization Cost. In our approach, we need to generalize vertex labels. The generalization cost is

$$GenCost(l_u, l_t) = \frac{|h_{l_t} - h_{l_u}|}{|GTree|} \tag{1}$$

l_u represents the original label of vertex u. l_t represents the target label of vertex u. h_{l_t} represents the height of label l_t in the generalization tree. h_{l_u} represents the height of label l_u in the generalization tree. $|GTree|$ represents the total height of the generalization tree.

Definition 4 Anonymity Cost. The cost of anonymizing $G = (V, E, L)$ to $G' = (V', E', L')$ is

$$Cost(G, G') = |E' \backslash E| + |E \backslash E'| + \sum_{v_1}^{v_m} GenCost(l_u, l_t) \tag{2}$$

Suppose there are m vertices in a graph.

4 LP k^2-Anonymity Approach

In this section, we devise two effective algorithms, one is LGA (Label Generalization Anonymization) for grouping the vertices and generalizing the labels of vertices. Another is used for anonymizing the social network graph by adding and deleting edges called LGAN (Label Group ANonymization). The two algorithms share the same purpose that is to preserve the utility while satisfying the LP k^2-anonymity.

4.1 Label Generalization Anonymization (LGA) Algorithm

In this section, we organize vertices into groups and generalize vertices' labels by Algorithm LGA. We require there exist at least k vertices in each group. And generalizing vertices' labels makes all the vertices in each group have the same label.

To get the goal as described above, first we should sort the vertices sequence f in a specific order. Take the order of subtrees into account. At first, we consider the first subtree. We scan the labels of vertices in the graph in a breadth-first way, and sort them by the order in which the vertices have the same label or have the label of sibling relationship. Then we handle the vertices with the labels in the following subtrees in the same way. And these vertices which are handled newly will be added into f. The process is finished until all the subtrees are considered, that is all vertices are sorted.

Suppose the sequence is $f = (v_1, v_2, v_3 \ldots v_m)$. Then we group the vertices into $GP_1, GP_2 \ldots GP_n$ and make sure that there exist at least k vertices in each group, there are m vertices in the graph. These m vertices should be divided into multiple groups. First, we put k vertices into GP, if $|GP| \geq k$, we should analyze v_{k+1} and v_k two vertices' label relationship. If they have the same label or have the label of sibling relationship, we put v_{k+1} into GP. Otherwise, we start another group. We mark $GR = \{GP_1, GP_2 \ldots GP_n\}$. Then we choose a target label for vertices in each group. The selection rule is made according to the smallest generalization cost in Eq. (1) introduced in Sect. 2.

Example 1. We take an example of LP 2^2-anonymity. First, we form a generalization tree for the social network graph which is shown in Fig. 4(a). The generalization tree is displayed in Fig. 3. The vertices in Fig. 4(a) are sorted as $f = \{3, 2, 4, 6, 1, 5, 7, 8\}$. We group them into three groups. $GP_1 = \{3, 2, 4, 6\}$, $GP_2 = \{1, 5\}$,

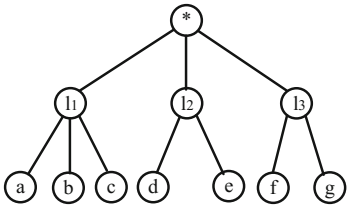
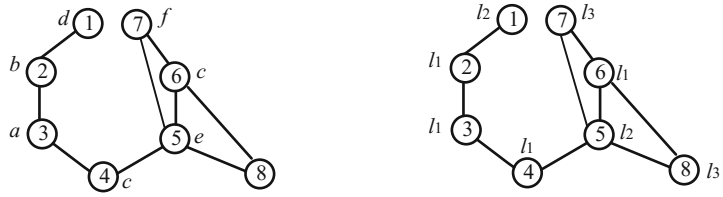


Fig. 3. An example of a generalization tree (GTree)



(a) original social network graph (b) label generalization graph

Fig. 4. Examples of generalizing labels

$GP_3 = \{7, 8\}$, $GR = \{GP_1, GP_2, GP_3\}$. We choose a target label l_1 for GP_1, l_2 for GP_2, l_3 for GP_3 and generalize the labels of vertices to be their target labels. The generalization result is shown as Fig. 4(b).

4.2 Algorithm Label Group Anonymization (LGAN)

We propose an algorithm named LGAN to add edges or delete edges of the vertices after grouping. First, for every vertex in each group, we examine whether there are not less than $k - 1$ other vertices owning the same label pair of (m, n) between groups. The group which satisfies the condition is removed from the set GR. The left groups in the set GR will be processed. The average degree of vertices in each group is calculated. The group with the highest average degree is selected firstly, and the process of adding edges or deleting edges is performed to ensure that each label pair (m, n) in the group is either zero or not less than k. We use table $VerTbl[x][y]$ to store the number of vertices in x with edges connecting to the vertices in y, x represents the vertices whose labels are m and y represents the vertices whose labels are n. $EdgTbl[x][y]$ stores the number of edges connecting vertices in x and y, and x represents the vertices whose labels are m and y represents the vertices whose labels are n. We ensure that two groups have enough edges by adding edges or deleting edges. For each label (m, n), if $0 < VerTbl[x][y] < k$ or $0 < VerTbl[y][x] < k$, we get the cost of edge addition, i.e., $k - min(VerTbl[x][y], VerTbl[y][x])$ and the cost of edge deletion, i.e., $EdgTbl[x][y]$. If the cost of edge deletion is no more than the cost of edge addition, we delete the edges between group x and group y. Then we set $VerTbl[x][y]$, $VerTbl[y][x]$, $EdgTbl[x][y]$, $EdgTbl[y][x]$ as zero. Otherwise, we add edges between group x and group y according the following strategy:

(1) vertex u (or v) in group x has no connection with v (or u) in group y;
(2) the shortest path between every two candidate vertices is the minimal one in the original graph.

After the group is handled, we remove the group from the set GR. We iterate this step until the set GR is empty.

Example 2. After Example 1, we get the graph of Fig. 4(b). On the basis of Fig. 4(b), we add edges or delete edges between groups. In Fig. 4(b), for every vertex in GP_3, there are not less than $k - 1$ other vertices owning the same label pair of (m, n) between groups. The group is removed from the set GR directly. We consider the left groups GP_1, GP-$_2$. Vertex 6 in GP_1 can be uniquely identified by the label pair (l_1, l_3). Similar, vertex 5 in GP_2 can be uniquely identified by the label pair (l_2, l_3). We calculate the average of the vertices in each group, we can know the average degree of GP_1 is 2.25, and the average degree of GP_2 is 2.5. As a consequence, we give priority to deal with the vertex 5 in GP_2. LGAN deletes an edge (5, 7) as shown in Fig. 5(a). Then, we remove GP_2 from the set GR. To protect vertex 6, an edge (4, 8) is chosen to add. Then, we remove GP_1 from the set GR. The set GR is empty that means the social network graph in Fig. 4(a) is already anonymized completely. Finally, the Fig. 5(a) is the LP 2^2 anonymous resulting graph.

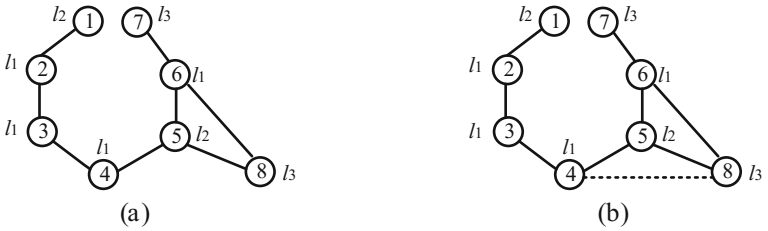


Fig. 5. An example of LP 2^2-anonymity graphs

5 Experimental Evaluation

In the section, we introduce the datasets and evaluate our algorithm. All the experiments are conducted in a virtual machine on a PC computer. The PC is with a 2.50 GHz Intel (R) Core(TM) i7-6500U CPU and 4.0 GB memory. The virtual machine runs Fedora release 23 system with 1.5 GB memory. The program is implemented in C.

5.1 Data Sets

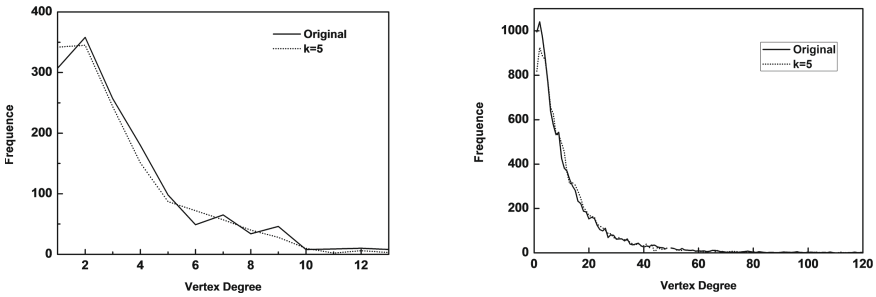
We conduct our experiments on two real datasets. One dataset is a co-authorship data in network science [24]. We construct a social network from the data and extract author names as labels. Each vertex in the graph represents an author, and two vertices are linked by an edge if the two corresponding authors co-authored at least one paper in the data set. There are 1461 vertices and 2742 edges in the co-authorship graph after removing the isolated vertices and the average degree is about 3.76. Another is from the e-print arXiv. We derive a graph describing the citations between papers from Arxiv HEP-TH (high energy physics theory) [25]. If one paper cites another paper, an undirected edge will connect both corresponding vertices. The graph includes 12130 vertices and 76043 edges after removing the isolated vertices. The average degree of vertices is about 12.54.

5.2 Data Utility

We evaluate the performances the LGA and LGAN algorithm by measuring the degree distribution, average clustering coefficient, average path length, the numbers of edge changes and running time.

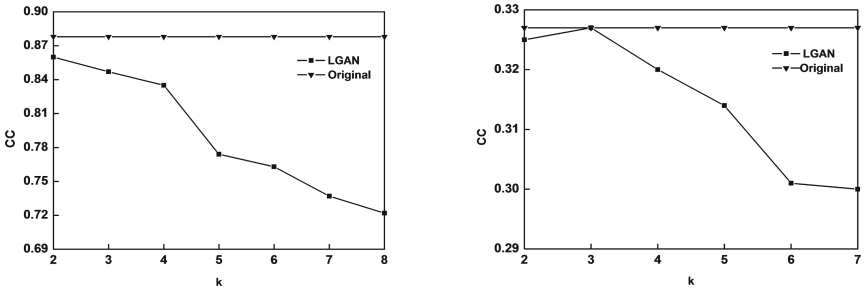
Degree Distribution. Figure 6 shows the degree distributions of the original graphs and the anonymized graphs. It can be seen the degree distributions of anonymized graphs are similar with the original graphs.

Average Clustering Coefficient (CC). Figure 7 compares the average clustering coefficients of the original graphs and the anonymized graphs. The basic trend is that the CC values on two datasets decrease when k increases. Specially, when k value is 3, the CC value of the HEP-TH dataset increases a little.



(a) network science dataset (b) HEP-TH data

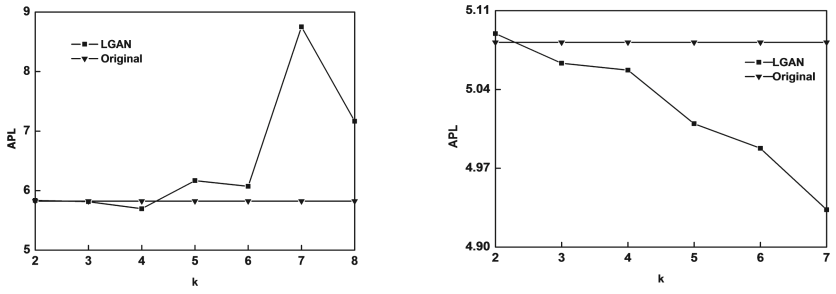
Fig. 6. Degree distribution



(a) network science dataset (b) HEP-TH dataset

Fig. 7. Average clustering coefficient

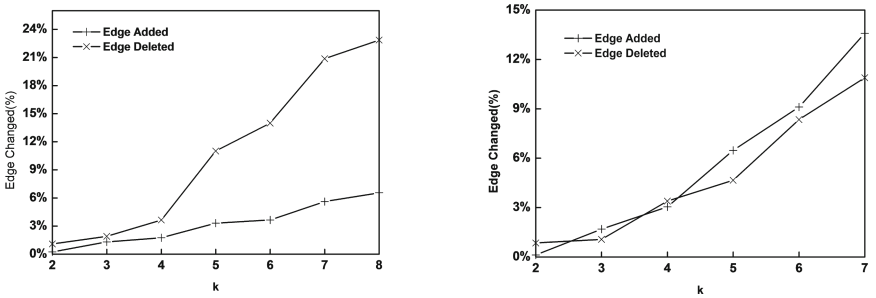
Average Path Length (APL). The average path lengths on two datasets for the original graphs and the anonymized graphs are shown in Fig. 8. The APL of the graph anonymized is very close to the APL of the original graphs, especially when k value is small.



(a) network science dataset (b) HEP-TH dataset

Fig. 8. Average path lengths

Percentages of Edges Changed. We consider the edge changes in our algorithm. Figure 9 shows the edge changes on the original graphs. The changes include the ratios of edges added and edges deleted. In our algorithm, we change the fewest edges. In the HEP-TH dataset, the vertex degree is smaller relatively than the dataset in the same size. For better experimental results, we make many vertices own the same label by generalizing labels. Also k value is set little when we perform the experiment.



(a) network science dataset (b) HEP-TH dataset

Fig. 9. Percentages of edges added and deleted

From the above evaluation, it can be seen our algorithm can preserve the utility of the original graph effectively.

Running Time. Figure 10 shows the runtime on the network science dataset with respect to different k values. We can know the runtime increases when the k value increases from the figure.

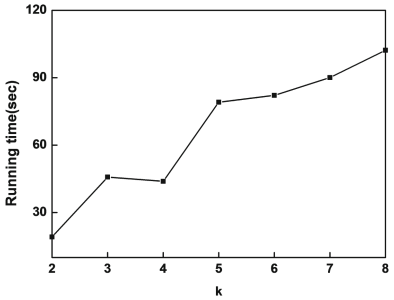


Fig. 10. The runtime on network science dataset

6 Conclusions

In this paper, we have proposed a new concept LP k^2-anonymity to protect individual privacy against a new type attack, called label pair attack. For LP k^2-anonymity, we provide a new method to anonymize the social graphs by algorithm LGA and LGAN.

In order to preserve the original graphs, we generalize the vertex labels as fewer as possible and only add necessary edges to construct a new graph without adding the noisy vertices in our algorithms. We also give a detail analysis of the data utility. The experimental results on two real data sets demonstrate that our approaches can preserve much of the utility of the original graph.

Acknowledgement. This work is supported by National Natural Science Foundation of China under Grant 61572459, 61672180 and 61602129. The paper is funded by the International Exchange Program of Harbin Engineering University for Innovation-oriented Talents Cultivation.

References

1. Cai, Z., He, Z., Guan, X., Li, Y.: Collective data-sanitization for preventing sensitive information inference attacks in social networks, p. 1 (2016)
2. Tai, C.H., Yu, P.S., Yang, D.N., Chen, M.S.: Privacy preserving social network publication against friendship attacks. In: Proceedings of KDD, San Diego, CA, pp. 1262–1270 (2011)
3. Campan, A., Truta, T., Cooper, N.: P-sensitive k-anonymity with generalization constraints. Trans. Data Priv. **2**, 65–89 (2010)
4. Fung, B.C.M., Wang, K., Yu, P.S.: Top-down specialization for information and privacy preservation. In: International Conference on Data Engineering, pp. 205–216 (2005)
5. He, Z., Cai, Z., Han, Q., Tong, W., Sun, L., Li, Y.: An energy efficient privacy-preserving content sharing scheme in mobile social networks. Pers. Ubiquit. Comput. **20**(5), 833–846 (2016)
6. He, Z., Cai, Z., Sun, Y., Li, Y., Cheng, X.: Customized privacy preserving for inherent data and latent data. Pers. Ubiquit. Comput. **21**(1), 1–12 (2016)
7. Hay, M., Miklau, G., Jensen, D., Towsley, D., Weis, P.: Resisting structural re-identification in anonymized social networks. VLDB J. **19**(6), 797–823 (2010)
8. Liu, K., Terzi, E.: Towards identity anonymization on graphs. In: Proceedings of SIGMOD, Vancouver, BC, pp. 93–106 (2008)
9. Liu, L., Wang, J., Liu, J., Zhang, J.: Privacy preserving in social networks against sensitive edge disclosure (2008)
10. Liu, X., Yang, X.: Protecting sensitive relationships against inference attacks in social networks. In: Lee, S.-g., Peng, Z., Zhou, X., Moon, Y.-S., Unland, R., Yoo, J. (eds.) DASFAA 2012. LNCS, vol. 7238, pp. 335–350. Springer, Heidelberg (2012). doi:10.1007/978-3-642-29038-1_25
11. Sun, C., Yu, P., Kong, X., Fu, Y.: Privacy preserving social network publication against mutual friend attacks. Trans. Data Priv. **7**, 71–97 (2013)
12. Wang, K., Yu, P.S., Chakraborty, S.: Bottom-up generalization: a data mining solution to privacy protection. In: ICDM, pp. 249–256 (2004)
13. Yuan, M., Chen, L., Yu, P.: Personalized privacy protection in social networks. VLDB **4**, 141–150 (2010)
14. Yuan, M., Chen, L., Yu, P., Yu, T.: Protecting sensitive labels in social network data anonymization. IEEE Trans. Knowl. Data Eng. **25**, 633–647 (2013)
15. Zheng, X., Cai, Z., Li, J.Z., Gao, H.: Location-privacy-aware review publication mechanism for local business service systems. In: The 36th Annual IEEE International Conference on Computer Communications (2017)

16. Zou, L., Chen, L., Zsu, M.T.: K-automorphism: a general framework for privacy preserving network publication. Proc. VLDB Endow. **2**(1), 946–957 (2009)
17. Zhang, L., Cai, Z., Wang, X.: Fakemask: a novel privacy preserving approach for smartphones. IEEE Trans. Netw. Serv. Manag. **13**(2), 1 (2016)
18. Zhou, B., Pei, J.: Preserving privacy in social networks against neighborhood attacks. In: ICDE, pp. 506–515 (2008)
19. Zhou, B., Pei, J., Luk, W.S.: A brief survey on anonymization techniques for privacy preserving publishing of social network data. ACM SIGKDD Explor. Newsl. **10**(2), 12–22 (2008)
20. Zheng, X., Cai, Z., Yu, J.Z, Wang, C.K., Li, Y.S.: Follow but no track privacy preserved profile publishing in cyber-physical social systems. IEEE Internet Things (2017)
21. Zhang, J., Sun, J., Zhang, R., Zhang, Y., Hu, X.: Privacy-preserving social media data publishing (2017)
22. Zhang, L., Zhang, W.: Edge anonymity in social network graphs. In: International Conference on Computational Science and Engineering, pp. 1–8 (2009)
23. Zhang, L., Wang, X., Lu, J., Li, P., Cai, Z.: An efficient privacy preserving data aggregation approach for mobile sensing. Secur. Commun. Netw. **9**(16), 3844–3853 (2016)
24. The Web Environment at U-M. http://www-personal.umich.edu
25. Stanford Network Analysis Project. https://snap.stanford.edu