De-anonymizing Social Networks: Using User Interest as a side-channel

Shuying Lai, Huaxin Li, Haojin Zhu, Na Ruan
Shanghai Jiao Tong University, Shanghai, China
{laisycs,lihuaxin0033,zhuhaojin,ruannana}@gmail.com

***Abstract*—Social networks, such as Twitter, Instagram, Facebook, and Weibo, have covered a wide range of population throughout the world. People use multiple social networks to enjoy various services as well as share their personal information according to different privacy level. Unlike Facebook and Wechat, in which social connection is more like acquaintance, social networks like Twitter, Instagram, and Weibo represent the social networks in which social link represents interest rather than acquaintance. According to this observation, we propose a community detection method based on interest group, then apply de-anonymization algorithm based on this community. Our experiment shows that this achieves better accuracy than existing de-anonymization algorithm. We conduct several experiments to demonstrate the effect of parameters used in the de-anonymization algorithm.**

I. Introduction

According to Social Media Update 2014 done by PewResearch Center [1], [2], 71% of adult internet users or 58% of entire adult population use Facebook, 23% of online adults use Twitter, 26% of online adults use Instagram. People use social media sites with multiple purposes. Social medias foster the communication both publicly through comments and likes or private messages. People build up connection, make friends, enhance relationship with others. Furthermore, people participate activities, do propaganda, find jobs based on the social relationships.

However, social network, promoting exposure of ordinary people, has become a significant source of privacy threat [3]. People intentionally or unintentionally reveal sensitive information to build personal profile on social networks [4], [5]. Usage of that information can fall into three categories. Firstly, it is used for academic and government data-mining. For example, professors in universities and experts working in the government collect data about friendship, health data etc for multiple research or data-mining purposes. Secondly, data is used for advertising. Companies advertising department pay for online social network operators to show personalised promotion and advertisement to users. Thirdly, people use data for third-party applications. To make registration and login process easier for users, a lot of application operators support login with online social network account. In most cases, those applications are granted the privilege to access users' personal profiles and friendships.

The same piece of social research done by PewResearch Center [1] shows that 52% of online adults use multiple social media sites at the same time. Due to the fact that social media sites have different focuses, people share different information on different social media sites and thus people have different privacy preference in different social media sites. By de-anonymizing users among different social networks, privacy of users will be exposed to a larger extent. Previous works have proposed various approaches to 'de-anonymize' users [6]–[8].

In this paper, we propose an enhanced version of de-anonymization algorithm to attack social network privacy and conduct a large-scale and passive de-anonymization experiment to demonstrate how factors influence the result of de-anonymization. Our main contributions are summarized as follows.

- First, we discuss the drawbacks of both community detection algorithm and applying community information in de-anonymization in previous work. We thus define a different scenario in which community should be detected not solely through network structure but through the semantics of social networks, meaning that divide users in social networks according to their interests inferred from information they reveal on social networks.
- Second, we demonstrate the effectiveness of our algorithm compared to basic community-blind de-anonymization algorithm. We apply our algorithm to two subgraphs of Instagram, a real-world large-scale social networks. Additionally, we conduct extensive experiments to analyse factors that affect the results of de-anonymization.
- Third, we discuss the possibility to extend this scenario and apply this enhanced de-anonymization algorithm to achieve better de-anonymization results.

The rest of paper is organized as follows. Section II introduces related research works. Section III explains motivations of our works. Section IV gives a formal definition of our problem. Section V presents our enhanced de-anonymization algorithm in details. In section VI, we evaluate effectiveness on real-world datasets. Section VIII concludes this paper.

II. Related work

Zhou and Pei [9], [10] propose privacy-preserving method against neighborhood attacks, they identify the essential type of privacy attacks: neighborhood attacks. The attackers are assumed to know exactly 1-neighbourhood of target users. Thus the attackers identify target users through its unique, as is assumed, neighbourhood structure. Tripathy and Panda [11] propose an modification of Zhou and Pei's algorithm.

978-1-5090-0243-6/15/$31.00 ©2015 IEEE

Instead of using Deepest First Search (DFS) method, they use an algorithm for graph isomorphism based on an graph representation-adjacency matrix. In such way they reduce the time requirement of the algorithm compared to Zhou and Pei's algorithm.

Narayanan and Shmatikov [6] presented an de-anonymization algorithm (NS algorithm) to map users in a target social network to users in an auxiliary social network. The auxiliary social network is assumed to have mutual users with the target social network. By mapping users in two networks, users in the target social network are totally de-anonymised if the auxiliary social network is not anonymised, or more users' information will be found out if the auxiliary social network is anonymised. This algorithm makes use of graph structure of social networks, which is represented by undirected graphs. Nilizadeh, Kapadia, and Ann presented an community-enhanced version of NS algorithm [7]. The algorithm shares basic idea and some steps with NS algorithm while it makes use of the community structure of social networks.

III. Motivation

A. Community detection based on graph structure.

Random walk. Current community detection methods are mainly based on graph structure of a graph, for example Infomap. Rosvall and Bergstrom [12] propose a community detection algorithm. They use probability path of random walks [13] on the graph and divide a graph into communities.

We argue that this method is only suitable for social networks containing strong social ties among users, such as Facebook, LinkedIn, Renren, or Wechat, where a person needs another person's confirmation to construct friendship. As a result, in these kind of social networks, people connect to who they know or familiar with and social network graphs have strong communities. However, this method is not effective for social networks like Twitter or Instagram because they do not have those features.

Community based merely on structure is not effective. It is reasonable to speculate that Instagram, a social media site which has almost exact identical structure as Twitter does, has similar features as Twitter does in above three aspects. Then, as users in Twitter and Instagram interact relatively randomly, use methods like random walk to detect communities will result in more significant deviation because the algorithm is more likely to 'walk along' a path on which nodes have weak connections to each other. Consequently, community detection based merely on graph structure does not serve social networks like Instagram and Twitter well.

B. Interest group

As is stated above, we discover that graph structure is not sufficient to handle community detection well enough. An observation is that a person does not belong to only one community if the size of the community is small. We introduce interest group into community detection. Our Intuition is that people have certain interests and they show those interest in social networks by following popular accounts that are related to their interests. Interest is defined by the semantics of an account. For example, followers of Twitter account 'BBS Sports' and 'SkySportsNewsHQ' can be grouped into interest group with label 'sports'. Interest can be either general or specific. Definition of interest decides the size of interest groups. There is no universal method to define interest groups. We state that this can only be done according to the conditions of target social network, including its size, characteristics etc.

IV. Problem definition

As is described before, since people reveal different information on different social media sites, one major de-anonymization attack is to map users in different sites. If a user's social network accounts are successfully associated, the user's privacy is breached because information is known to people who are not supposed to know.

As our target social network has both 'follow' and 'friend' relationship between two users, we model a social network as $G = (V, E, \mathcal{V}, \mathcal{E})$, where V is a set of vertices, $E \subseteq V \times V$ is a set of edges, $\mathcal{V}$ is the attribute sets of vertices V, and $\mathcal{E}$ is the attribute sets of edges E. The most important attribute of an edge is direction which illustrates the relationship of the users. We define $e(v_1, v_2), e \in E, v_1, v_2 \in V$ means that v_1 follows v_2, v_1 is a follower of v_2, and v_2 is a friend of v_1.

In our problem, we assume the attackers know their target social network graph S_{tar}. In addition to S_{tar}, the attackers have access to an auxiliary social network graph S_{aux} which have some overlapped users with target graph S_{tar}.

Our problem is to map S_{tar} to S_{aux} social networks. The attackers are assumed to have access to a graph which represents a social network after edge addition and deletion. In this graph, existence of nodes and edges, i.e. graph structure V, E, are obviously known to attackers. Different from graph structure, information inside edges or nodes such as edge attributes, nodes attributes etc. can not be accessed by attackers. An exception is that the only edge attribute that the attackers know is the direction of edges.

Privacy protection model. In order to protect the users' privacy before social network $S_0 : G_0 < V_0, E_0 >$ is published, we remove users' identification in V_0. Additionally, we conduct random edge modification to add noise to S_0 with noise degree θ. Noise degree θ is the number of operations of edge moderation, including random edge addition and edge deletion. The noise-degree-added graph is the published social network as well as the target of the attacker S_{tar}.

Attack model. The attacker is assumed to have access to target graph $S_{tar} : G_1 < V_1, E_1 >$ and public auxiliary graph $S_{aux} : G_2 < V_2, E_2 >$. Also, the attacker has knowledge of top k popular accounts in V_1, which means the attacker knows who those users are or what those accounts about. The attacker's goal is to find a function:

$f : V' \to V_2, V' \subseteq V_1, f$ is an injection but not a surjection. (1)

V. Methodology

A. Interest group identification

Interest group identification is a heuristic approach. It takes a graph of social network as input and outputs the graph divided into groups in which users have common interest.

Power user identification. First of all, power users are identified from both graphs. Those popular users can be easily found according to their extraordinarily large number of incoming edges. Normally, top 1% of most popular users are enough to be used in interest group identification.

Assign interest tag to power users. Secondly, an $interest\ tag$ is assigned to every power user to illustrate their interest attribute. This step only takes negligible amount of manual effort because power users are popular and famous in some area, and they all have a clear and specific group of followers. Note that one user can be assigned to several interest tag and that an interest tag can be assigned to more than one users.

Detect interest group. Interest groups are detected with intuition that followers of a power user belong to a group with the interest tag corresponding to the power user. If a user belongs to more than one interest groups, a majority vote is conducted to decide which interest group the user is belonged to.

We create $Label()$ function to assign power user a label to represent the interest group. This function has no precise implementation. In our experiment, we use our manual work to replace $Label()$ function. The interest group detection procedure is as follows:

Algorithm 1 Interest group detection

Input A graph $G < V, E >$, number of interest group n
Output A dictionary $groups$ in which keys are interest group labels, values are lists of power users that belong to the interest group labels.

```
group={}
while true do
    Find the most popular user U_power account in G
    label=Label(U_power)
    if label not in group.keys() then
        group[label]=[U_power]
    else
        group[label].append(U_power)
    end if
    if number of labels in group == n then
        break
    else
        G = G − U_power
    end if
end while
```

B. Propagation

In this section, we describe the propagation step of our algorithm. The propagation makes use of graph structure as well as interest group structure to propagate the seed mapping to the whole graph. We traverse the already mapped node pairs and search for more mappings between their neighbours.

For each pair of mapped node $u_1 \in S_{tar}$, $u_2 \in S_{aux}$, we find all their neighbours $N_1 \in S_{tar}$, $N_2 \in S_{aux}$. For each $n_1 \in N_1$, we evaluate the extent that n_1's neighbours are mapped to n_2's neighbours, where $n_2 \in N_2$. We evaluate it by calculating scores for n_1 and each $n_2 \in N_2$. More precisely, if we discover one of n_1's neighbour has already been mapped to one of n_2's neighbour, we plus 1 to the score of n_2. After the evaluation is done, we pick the node in N_2 with the largest score for n_1. Finally we add the new mapping to existing mapping we have.

Interest group. When we are evaluating the similarity of two potentially mapped nodes, we only consider the extent that two nodes' neighbours are mapped. However, the attribute of the two potentially mapped nodes should be considered important for the following reasons. First, it is straightforward that similar interest implies general similarity between two nodes. Though people use multiple social media sites for different reasons, their interest revealed in each social media site is unlikely to have considerable inconformity. Second, there is no reason to believe S_{tar} is the original social network data without sanitation. Actually, published social networks have in most cases been through a privacy protection processing including noise addition, edge change etc. However, to preserve the utility of published data, this kind of processing is minor compared with the size of published dataset and can not change the basic interest to which a user belong. So, considering the interest group of the nodes themselves helps us rule out disturb from noise.

Eccentricity. When we pick the node with largest score, we remove all candidate nodes with a smaller score. However, the result may not be desirable if the false node get largest score while the true mapping node get a score that is slightly smaller than the largest score but still obvious larger than other scores. Here, we need to value the eccentricity of the candidate set to ensure that this node is the only one node, one with the largest score, that stands out in the candidate set. The eccentricity of a set X is defined as follows:

$$\frac{max(X) - max_2(X)}{\sigma(X)}$$

where $max_2(X)$ is the second largest value in X. We set a threshold θ for eccentricity of score set. If the eccentricity of score set is smaller than θ, we reject the mapping and continue. This rules out the potential noise and ensures the accuracy of mapping.

Reverse match. Reverse match is needed when we have found one node in N_2 for n_1. At this stage, we have only found the best match n_2 for n_1 among nodes in N_2. However, we can not make sure whether reversely, n_1 is the best match

for n_2 in N_1. Therefore, the next step is to find reverse match-the best match for n_2 in N_1. If the reverse best match is n_1, we guarantee the correctness of mapping n_1 and n_2 in both directions and accept the mapping. Otherwise, we reject the mapping and continue.

Node degree. We found that the propagation step is in favour of nodes with large number of neighbours. Here is an example to illustrate this discovery. Assume that there are two nodes in N_2, one have 80% of its neighbours mapped to the neighbours of n_1 while the other have 10% of its neighbours mapped to the neighbours of n_1. It is straightforward that the former should get a higher score than the latter because its already mapped neighbour has a much larger coverage. However, if the latter happens to be a node with 10 times more neighbours than the former has, the latter will get a higher score in spite of low coverage. Our approach is that when we find a pair of mapped neighbours for n_1 and n_2, we plus $1/degree(n_2)$ to the score of n_2. This makes the algorithm detached from node degree.

We present our de-anonymization algorithm in Algorithm 2.

Algorithm 2 De-anonymization

Input $G_1 < V_1, E_1 >$, $G_2 < V_2, E_2 >$, seed mappings
Output $\{< v_1 : v_2 > | v_1 \in V_1, v_2 \in V_2\}$
  mapping=add seed mappings
  while not conversion **do**
    for $< v_1, v_2 > \in mapping$ **do**
      PROPAGATE(v_1, v_2 , G_1, G_2)
    end for
  end while

VI. Experiment setup

A. Data sets

We collected user relationship data from Instagram, a popular photo-sharing social network. We build a graph where nodes represent users and edges represent follow and friend relationships. Since relationship has two types and there is no attribute considered in our main part of the algorithm, our graphs are all directed unweighted graphs.

Statistics of graphs are shown in Table. I. In our large scale experiment, S_{tar} has as many as 823 thousands nodes and one million edges. The average node degree is 1.44. S_{aux} has as many as five million nodes and nine million edges. The average node degree is 1.64. In our small scale experiment, S_{tar} has as many as 18 thousands nodes and 30 thousands edges. The average node degree is 1.78. S_{aux} has as many as 25 thousands nodes and 40 thousands edges. The average node degree is 1.62.

B. Noise addition

We realize that if two accounts in the two graphs belong to the same user, their will have exactly the same graph structure. This makes our experiment hypothetical and fail to evaluate the true performance of our algorithm in the real world. To emulate real world scenario, we add around 10% random noise to both of the social networks to make the two social networks more like two networks operated different operators.

OSN	Username	Number of nodes	Number of edges	Average degree
Instagram	Columbia theta	$823K$	$1M$	1.44
Instagram	Columbia	$5M$	$9M$	1.64
Instagram	Columbia sce	$54K$	$71K$	1.5
Instagram	mrkevinmullen	$18K$	$30K$	1.78

TABLE I: Graph statistics

Algorithm 3 Noise addition

Input $G_1 < V_1, E_1 >$ and a noise degree parameter θ
Output $G_2 < V_2, E_2 >$ where $V_1 = V_2$ but $E_1 \neq E_2$
  $G_{=}G_2$
  $n = 0$
  while $n < \theta$ **do**
    operation=randomly choose between add or delete
    randomly choose a node v_1
    if operation==add **then**
      randomly choose v_2 where $v_2 \in V_1$ but $< v_1, v_2 > \notin E_2$
      $E_2 = E_2 \cup < v_1, v_2 >$
    end if
    if operation==delete **then**
      randomly choose v_2 where $< v_1, v_2 > \in E_2$
      $E_2 = E_2 - < v_1, v_2 >$
    end if
    $n+ = 1$
  end while

In our experiment, we use the following value of noise: $\{0.01, 0.1, 0.2\}$

C. Experiment Setting

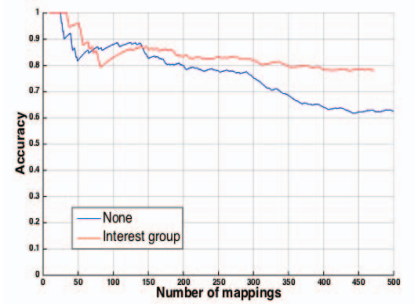
Seed selection. Due to the limitation of computation, we select seed manually by finding seeds with same Instagram ID. If the experiment is conducted on two social networks powered by different operators, we check the username and full name of users and pick out user pairs that have small Levenshtein distance.

Eccentricity. In our experiment, we use the following value of eccentricity: $\{0.1, 0.2, 1, 1.5, 2\}$. For each eccentricity value, we draw the accuracy plot line and compare their impact on accuracy.

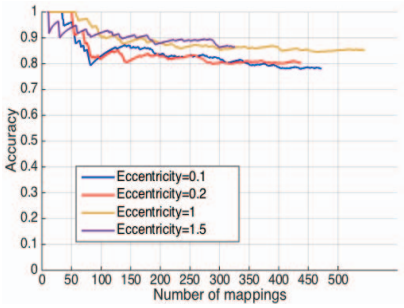
VII. Performance Evaluation

We perform experimental evaluation on the large scale Instagram dataset.

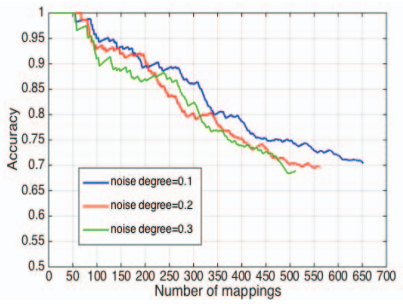
1) Accuracy enhancement: As is shown in Figure 1(a), the accuracy of our interest-group-enhanced algorithm outperforms the original NS algorithm 20 percent when the algorithm propagates to the stage at which the size of the mapping M is 450. Due to the limitation of computational resources and



(a) Accuracy comparison between original NS algorithm and our interest group enhancement



(b) Accuracy under different eccentricities



(c) Accuracy under three noise degree levels

Fig. 1: Experimental evaluations

time resources, we are not able to record the final accuracy when the algorithm converges. To avoid eccentricity and noise degree parameters to have a significant impact on the result of accuracy, we give both the interest-group-enhance algorithm and original NS algorithm eccentricity value 0.1 and absolute noise degree value 1000 for S_{tar} graph of Columbia theta and 10000 for S_{aux} graph of Columbia university.

2) Accuracy analysis based on eccentricity value.: When we are studying the impact that eccentricity value has on accuracy of mapping in the process of propagation, we control the value of noise degree to a fixed value.

Eccentricity	Mapping M size	Accuracy
0.1	400	0.7875
0.2	400	0.8025
1	500	0.852
1.5	300	0.863

TABLE II: Accuracy with different eccentricity value.

As we can see in figure 1(b), eccentricity value has positive correlation with accuracy of the algorithm. The result values are also shown in Table. II. The reason causing this positive correlation is that higher eccentricity value guarantee the higher extent to which the node we pick with biggest matching score has outstandingly large matching score in the candidate group.

3) Accuracy analysis based on noise degree: In our experiment, we set the absolute noise degree value of Columbia graph and Columbia theta graph. Noise degree means the absolute value of noise degree (number of edge addition and deletion in total) in graph. As we can also see in Figure 1(c), the accuracy of our interest group de-anonymization algorithms decreases with the increase of noise degree. The plot line named noise degree 1 stays almost always on top of the plot line named noise degree 2. And the plot line named noise degree 2 stays almost always on top of the plot line named noise degree 3. The results indicate that adding noise will modify structure of graphs, thus reduces the accuracies.

VIII. Conclusion

In this paper, we define strong-community social networks and weak-community social networks. We propose an community detection method for weak-community social networks based on interest group, semantics of a graph, and then apply de-anonymization algorithm based on this division. The method we propose is based on users' following popular accounts or celebrities. To show effectiveness of our algorithm, we collect real-world Instagram and Twitter datasets. Our experiment shows that our method achieves better accuracy than original NS de-anonymization algorithm. Other experiments shows the the effect of parameters used in the de-anonymization algorithm.

Acknowledgment

This work is supported by National High-Tech R&D (863) Program (no. 2015AA01A707) and National Science Foundation of China (no. 61272444, U1401253, U1405251, 61411146001).

References

[1] Pew Research Center, "Social Media Site Usage 2014", 2015. http://www.pewinternet.org/2015/01/ 09/social-media-update-2014/.
[2] Pew Research Center, "Social Networking Fact Sheet", 2015. http://www.pewinternet.org/fact-sheets/ social-networking-fact-sheet/.
[3] Zhang C, Sun J, Zhu X, et al. "Privacy and security for online social networks: challenges and opportunities", *Network*, IEEE, 2010.
[4] M. Li, H. Zhu, Z. Gao, S. Chen, K. Ren, L. Yu, and S. Hu. "All Your Location are Belong to Us: Breaking Mobile Social Networks for Automated User Location Tracking," In *ACM MobiHoc'14*, 2014.
[5] Xiao, Q., Chen, J., Yu, L., Li, H., Zhu, H., Li, M., and Ren, K. "POSTER: LocMask: A Location Privacy Protection Framework in Android System." In *Proc. of CCS*, ACM, 2014.
[6] Narayanan A, Shmatikov V, "De-anonymizing social networks", *Security and Privacy*, IEEE, 2009.
[7] Nilizadeh S, Kapadia A, Ahn Y Y. "Community-enhanced de-anonymization of online social networks", *CCS*, ACM, 2014.
[8] Wondracek G, Holz T, Kirda E, et al. "A practical attack to de-anonymize social network users", *Security and Privacy*, IEEE, 2010.
[9] Ji S, Li W, Gong N Z, et al. "On Your Social Network De-anonymizablity: Quantification and Large Scale Evaluation with Seed Knowledge", 2015.
[10] Mickoleit A., Social Media Use by Governments: "A Policy Primer to Discuss Trends, Identify Policy Opportunities and Guide Decision Makers", *OECD Publishing*, 2014.
[11] Ding X, Zhang L, Wan Z, et al. "A Brief Survey on De-anonymization Attacks in Online Social Networks", *CASoN*, IEEE, 2010.
[12] Rosvall M, Bergstrom C T, "Maps of random walks on complex networks reveal community structure", *Proceedings of the National Academy of Sciences*, 2008.
[13] Wikipedia contributors, "Random walk," *Wikipedia, The Free Encyclopedia*, 2015. http://en.wikipedia.org/wiki/Random_walk.