Online link disclosure strategies for social networks

Younes Abid, Abdessamad Imine, Amedeo Napoli, Chedy Raïssi, Michaël Rusinowitch

Online link disclosure strategies for social networks*

Younes Abid, Abdessamad Imine, Amedeo Napoli, Chedy Raïssi and Michaël Rusinowitch

INRIA Nancy and Lorraine University, France
firstname.lastname@inria.fr

Abstract. While online social networks have become an important channel for social interactions, they also raise ethical and privacy issues. A well known fact is that social networks leak information, that may be sensitive, about users. However, performing accurate real world online privacy attacks in a reasonable time frame remains a challenging task. In this paper we address the problem of rapidly disclosing many friendship links using only legitimate queries (i.e., queries and tools provided by the targeted social network). Our study sheds new light on the intrinsic relation between communities (usually represented as groups) and friendships between individuals. To develop an efficient attack we analyzed group distributions, densities and visibility parameters from a large sample of a social network. By effectively exploring the target group network, our proposed algorithm is able to perform friendship and mutual-friend attacks along a strategy that minimizes the number of queries. The results of attacks performed on active Facebook profiles show that 5 different friendship links are disclosed in average for each single legitimate query in the best case.

Keywords: Online Social Network (OSN), Link disclosure attacks, Privacy

1 Introduction

A social network can be defined as a website that allows users to create personal pages in order to share information with their friends and acquaintances. These pages are usually called profiles and contain personal information. Profiles are connected to each other through friendship links that can be either symmetric or asymmetric, depending on the network's policy. Since their appearance at the end of the twentieth century, social networks have known an outstanding success and have become a global phenomenon. For instance, Facebook connects about 25% of humans in 2016[1] and YouTube served videos to almost one-third

* This work is funded by Fondation MAIF.

[1] http://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/

of all connected people on the Internet[2]. With this rapid network expansion, new scientific fields have emerged such as online social network analysis [8] creating a common domain of interest from sociology to mathematics and computer science [7]. However, the emergence of social networks is also giving reasons to worry about privacy and ethics issues [11].

In order to mimic real (i.e., non-cybernetical) societal interactions, some social networks such as Facebook, Linkedin and Viadeo support the creation of groups besides the profile creation. Accordingly, social networks can be modeled by two types of graphs – friendship graph and group membership graph – as depicted by Figure 1.

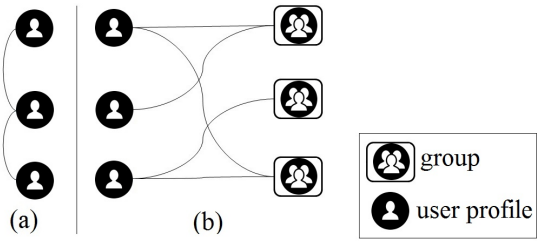

Fig. 1: Social graphs : (a) unipartite friendship graph, (b) bipartite group membership graph.

The friendship graph (a) is unipartite and models the friendship links between users while membership graph (b) is bipartite and models the membership links between users and groups. Some of these links can be masked by users or group administrators. We call a friendship (resp. membership) attack a sequence of actions (e.g., queries) leading to disclose a masked friendship (resp. membership) link. Both kind of attacks are called link disclosure attacks. A mutual-friend attack discloses common friends to a target and other users. We call group uncovering attack a sequence of queries that disclose the membership network of the target and his acquaintances. In this work the attacker is limited to the usage of legitimate and minimal queries provided by the social networks APIs. Therefore the attacker model can be viewed as a passive one. We believe that these constraints are the cornerstones of successful real-world attacks that are difficult to detect because the traffic appears to be legitimate at first.

In [3] researchers propose a Partial Graph Profile Inference (PGPI) algorithm that exploit group memberships to infer profiles attributes. In [10], relational learning approaches and group memberships are used to infer sensitive attribute of users such as locations.

Our experiments over 1,000 active Facebook profiles hiding their friend lists show that in the worst case 2 queries (in average) are sufficient to disclose at

[2] `https://www.youtube.com/yt/press/en/statistics.html`

least one friendship link and 5 different friendship links are disclosed in average by each query in the best case.

To put the rest of the paper into context, we start by defining problematics and objectives of link disclosure attacks on Online Social Networks in Section 2. Then we analyse groups distribution, densities and visibility parameters in Section 3. Those properties are then used to perform group uncovering attack as detailed in Section 4. In Section 5 we depict membership, friendship and mutual-friend attacks steps and we analyse their results. Finally, in Section 6 we give more detail about the resulting dataset of the attacks performed online.

2 Problematics and objectives

Problematics. In online attacks, the attacker is constrained by the network dynamicity and the time needed to scrap it. In fact, the dynamical network structure, with the addition/deletion of new links and nodes will ensure that the sampled graph does not reflect a real online social network at any given time. Therefore, crawling tasks for online attacks must be highly selective to collect only useful profiles and information and be as fast as possible.

For instance, [5] show that homophilic attributes have significant influence on predicting friendship between users of Facebook. Thus, an attacker may be tempted to sweep the network for similar profiles to his target. He can also consider the friends of the target friends as potential friends and check these links. Although these general solutions may seem effective to gather many potential friends, they have major shortcomings. To understand these shortcomings let us recall the "six degrees of separation" phenomenon, that is the possibility to connect any two people in a maximum of six relationship steps. For example, the authors of [1] show that the average degree of separation between Twitter users is 3.43 while the degree of separation on Facebook is between 2.9 and 4.2 for the majority of users [4]. Hence, considering friends of friends as potential friends is equivalent to considering at least tens of thousands users as potential friends for each single target [9]. This is clearly impossible to handle and scale for real-world efficient attacks.

Objectives. Link disclosure attacks in online social networks aim to disclose hidden links by performing authorized requests. The attacks either reveal existing links or potential ones according to the employed method. In this work we aim to improve the accuracy of the attacks. We aim to disclose numerous links without having to verify a huge number of potential friends. In other words, we attempt to gather many potential friends but only those who have high probability to be friend with the target. The best way to achieve our objectives is to disclose the vicinity network of the target. To that end, we analyse groups' properties on online social networks since they reflect the way users are gathering within a network and uncover its structure. To keep our discussion simple, we aim to answer two questions in this work: Which groups leak useful information to meet previously detailed objectives? And how to find and use them?

3 Social networks group properties

In this section we analyse some properties of Facebook groups. This analysis will guide crawling tasks in order to collect only data that leak more information about the target. Exploiting such data will increase the accuracy of link disclosure attacks and maximize the number of disclosed links. We stress that all experiments in this work were carried out online with real Facebook profiles. We have crawled 1,100 Facebook groups and all their members. Then, we have sorted the groups by declared size in sets. Each set contains at least 30 groups. Each group in the first set S_0 gathers between 2 and 10 members. And each group in the set S_i gathers between $10i$ and $10(i+1)$ members.

3.1 Group distribution

We first study the distribution of groups in Facebook with regard to their sizes. We notice that the declared group size on this network is often different from the number of users published on the group member list. Moreover, crawling the same group using different IP addresses and accounts can result in slightly different listed members. This technique can reduce the gap between the two sizes by considering the union of all crawled member lists of the same group. However, it adds more complexity to attacks. To study groups distribution we have simulated a simple attack carried out using only one attacker node. All groups are crawled only once and we only rely on the declared group size to build the attack strategy.

Figure 2 shows that there are many more small groups on Facebook than larger ones. However, we notice the curve inflection for groups declaring between 30 and 70 members. By checking these groups members lists we notice huge gaps between declared sizes and the numbers of listed members. Gaps reach 85% for some groups. Some groups are declared to have 60 members or more but they actually display less than 20 members on their members lists. These gaps can be explained by the fact that users unceasingly leave and join the group but size updates are not performed instantaneously. Henceforth, densities of such groups can increase if real sizes decrease since the less connected members are usually the first ones to leave the groups.

The result of our tests carried on 14,517 Facebook profiles shows that the probability of a given Facebook user to join at least one group gathering less than 50 members and publish his membership to it is 0.49. Thus, about half of analysed Facebook profiles are exposed to the danger of friendship link disclosure through groups they join and that gathers less than 50 members .

3.2 Group densities

In an undirected social graph, a friendship link between two user is considered public if at least one of them publishes it. It is considered hidden only if both users hide it.

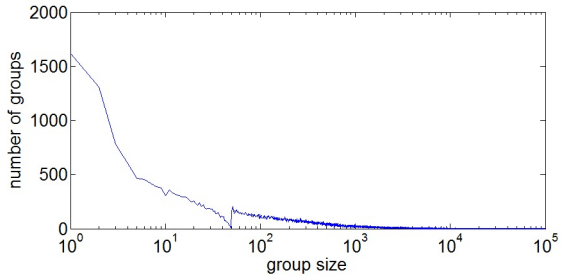

Fig. 2: Group distribution of a sample of 14,517 Facebook users.

In order to guide a strategy for disclosing hidden social links we first try to evaluate the probability that two members of a group are friends. We define three notions of group densities: public density, real density and maximal density, that we will use to estimate the number of friends that can be disclosed through link disclosure attacks. Given a group g, $PD(g)$ stands for its Public Density, $RD(g)$ stands for its Real Density and $MD(g)$ stands for its Maximal Density. The public density of g is the ratio of published friendship links between its members to all possible friendship links between them. It is defined by Equation (1) where $|g|$ is the number of members of g:

$$PD(g) = \frac{2}{|g|(|g|-1)} \sum_{\{m,m'\} \subseteq g} publicLink(m, m') \tag{1}$$

The real density of g is the ratio of all (public and hidden) friendship links between its members to all possible friendship links between them. It is greater or equal to the public density. It is defined by Equation (2)

$$RD(g) = PD(g) + \frac{2}{|g|(|g|-1)} \sum_{\{m,m'\} \subseteq g} hiddenLink(m, m') \tag{2}$$

The maximal density of g can be met only if all its members who hide their friend lists are friend with each other. It is greater or equal to the real density. It is defined by Equation (3) where p is the percentage of members who hide their friend lists among the members of g.

$$MD(g) = PD(g) + \frac{p^2|g| - p}{|g| - 1} \tag{3}$$

Thus we have:

$$PD(g) \leq RD(g) \leq MD(g) \tag{4}$$

Test results show that among 14,517 crawled Facebook profiles only 6,249 (43%) hide their friend lists or choose to reveal them only to their direct friends, friends of friends or some selected users. The rest (57%) leave the visibility

setting by default and publish their friend lists. Hence, p can be considered equal to 0.43 if it is unknown by the attacker. Note that the attacker can easily verify the friend list visibility parameters of other users through the following Facebook request:

$$/ < nid\_u > /friends \tag{5}$$

where nid_u is the numeric id [3] of the User u. In fact, this request returns the friend list of the User u if and only if he publishes it.

Figure 3 (a) shows that group densities decrease as the declared size of the group increases. It can be noticed that one can even estimate a given group density only from its declared size. This information is precious as it determines the number of links that can be disclosed between group members. In fact, the group real density can be viewed as the probability of the friendship link between a given member and another member from the same group. Hence, if the attacker discloses group membership of his Target t to a Group g, then all other members of g can be considered as potential friends of t with a probability in interval $[PD(g); MD(g)]$. Knowing the declared size of g, $PD(g)$ can be directly deduced from Figure 3 (a) and $MD(g)$ can be deduced from Equation (3). For instance, the average public density of groups gathering between 10 and 20 members is 0.343. Then, according to Equation (3) the real density of such groups belongs to interval $[0.343; 0.515]$ for p equal to 0.43. Expressively, the estimated accuracy of link disclosure attack is 0.343 and all the members of corresponding groups can be considered as potential friends with probability in $[0.343; 0.515]$.

Although popular groups gather many members, probabilities of friendship between them are very low. Crawling such groups is fruitful to seek a lot of potential friends of the target but with low probabilities. However, minute groups open small horizon for potential friends but with higher probability of friendship.

The relationship status between two members of a group g is a binary variable. Hence, assuming independance of friendship links in a first approximation, the expected number of published friendship links between a given member and all other members of the same group is the expectation of a binomial distribution of parameters $B(|g|, PD(g))$ which is $|g| \times PD(g)$. For example, Figure 3 (a) shows that the expected public density of groups gathering less than 11 members is greater than 35%. Hence, the expected number of friends of a target within a group he joins and that gather 6 members is 2 (since $0.35 \times 6 = 2.1$). Figure 3(b) shows that the expected number of disclosed links between the target and group members slightly increases as the declared size of groups increases. Note that x-axis unit correspond to 10 members and y-axis unit correspond to 1 friendship link.

3.3 Group visibility parameters

Groups and members can independently choose to publish or hide the membership relation. For instance Facebook users can choose to mask some groups from

[3] Numeric id can be acquired through `http://findmyfbid.in/`

their list of groups. On the other hand, the administrators of groups can independently publish the entire lists of members. With that in mind, an attacker can build an attack strategy to disclose the groups that are masked by users or the membership lists of secret groups.

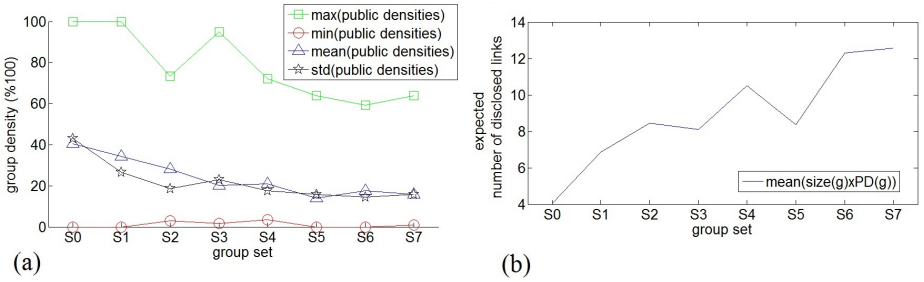

Fig. 3: Results of analysis: (a) Variation of public density with respect to group declared size, (b) Expected number of disclosed links between the target and group members.

4 Group uncovering attacks

In this section we exploit groups properties detailed in previous section to perform group uncovering attacks. To that end, we define real and public n-hop distant groups.

4.1 Real n-hop distant groups

Given a target t that joins Group g, g is considered as a real 1-hop distant group from t (denoted by $g \in RG_1(t)$) and all its members m are considered as real 1-hop distant members from t (denoted by $m \in RM_1(t)$). We define inductively $g \in RG_n(t)$ iff $g \notin RG_{n-1}(t)$ and there is $g' \in RG_{n-1}(t)$ with a non-empty intersection with g. For all m in $g \setminus RM_{n-1}(t)$ we have by definition $m \in RM_n(t)$. We can show the following symmetry rule:

$$u1 \in RM_n(u2) \iff u2 \in RM_n(u1) \tag{6}$$

where $u1$ and $u2$ are two different users. Figure 4 depicts an example of a real 3-hop distant group from the target node t.

Group $g1$ is a real 1-hop distant group from t. Consequently, all its members are real 1-hop distant members from t. Members $m6$, $m7$, $m8$, $m9$ and $m10$ are real 2-hop distant members from t since they join the same Group $g2$ as $m5$ who is real 1-hop distant members from t. Finally, $m11$, $m12$, $m13$ and $m14$ are real 3-hop distant members from t as $m9$ and $m10$ join their Group $g3$. Members $m5$, $m10$ and $m9$ act as gateway between groups.

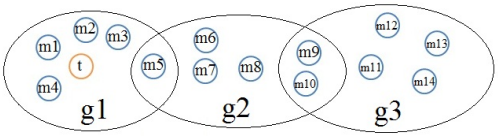

Fig. 4: g3 is a real 3-hop distant group from t.

4.2 Public n-hop distant groups

Users can mask their membership to groups and groups can hide their members lists. Consequently, the public n-hop distant relation does not satisfy the symmetry rule.

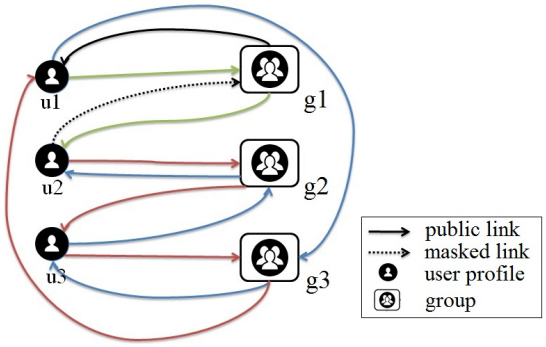

Fig. 5: An example of public n-hop distant groups and members.

Figure 5 depicts an example of different public n-hop distant groups and members between two users. Arrows from user to groups stand for membership links while arrows on the opposite direction represent group members lists. Dotted lines represent masked links and solid lines represent public links. While both Users $u1$ and $u2$ join the same group $g1$, only $u1$ publishes his membership to $g1$. User $u2$ publishes only his membership to Group $g2$. User $u3$ acts as a gateway between $g2$ and $g3$ and publishes his membership to both of them. All groups $g1$, $g2$ and $g3$ publish their member lists. There are two public paths from User $u1$ to User $u2$. The first one, the green path, goes through $g1$ and is the shortest one with only one hop. The second one, the blue path, is two hops long. It goes through $g3$ then $g2$. Hence, $u2$ is a public 1-hop distant member from $u1$. On the other hand, there is only one public path, the red path, from $u2$ to $u1$ that goes through $g2$ then $g3$. Thus, $u1$ is a public 2-hop distant member from $u2$.

4.3 Social graph traversal algorithm

Let $u2$ be a target user who is friend with both users $u1$ and $u3$ and hides his friend list. Since he publishes his membership to $g2$, the attacker can reach $u3$ through $g2$ member list. And if $u3$ publishes his friend list, the attacker can easily disclose the friendship link between $u3$ and $u2$ by checking the friend list of $u3$. Likewise, the attacker can reach $u1$ through $g3$ member list if $u3$ publishes his membership to that group. And he can search for $u2$ in $u1$ public friend list. Furthermore, next hop lead to $g1$ and hence the attacker can disclose group membership links between $u2$ and $g1$ by checking $g1$ public member list. Algorithm 1 gives more details about the graph traversal steps. The algorithm outputs are two sets of groups and members. And its inputs are the number of hops and a set of seed groups of the target.

```
   Data: gps: set of groups, h: number of hops
   Result: d_m: set of distant members, d_g: set of distant groups
 1 Procedure explore(gps, h, d_g, d_m)
 2   if (h > 0) then
 3     for each g ∈ gps do
 4       members.addAll(getMembers(g)) ;
 5     end
 6     members.removeAll(d_m);
 7     for each m ∈ members do
 8       groups.addAll(getPublicGroups(m)) ;
 9     end
10     groups.removeAll(d_g);
11     d_g.addAll(groups);
12     d_m.addAll(members);
13     explore(groups, h − 1, d_g, d_m);
14   end
15 Return()
```

Algorithm 1: Groups uncovering attack through social graph traversal

To collect seed groups, the attacker can directly retrieve unmasked groups from his target profile. We note that among 14,517 attacked Facebook profiles 11,446 (78.84%) do not change group visibility parameters and publish their groups membership even to secret groups. Otherwise, if the target masks all his groups and attributes, the attacker can create a fake virgin profile, use it to only visit his target profile, send him friendship request and try to interact with him by liking and commenting his posts or sending him messages. Then, link prediction algorithms of the social network [2] will start suggesting groups and attributes to the attacker that are strongly related to his target. Hence, he can use the suggested groups as seeds or take advantage of network research features and uses suggested attributes to look for seed groups. For instance, one of this

paper author hides all his attributes on Facebook. However, the social network suggested his home town and 10% of his friends to a newly created profile that he added as a friend.

By following Algorithm 1 steps the attacker can effectively crawl his target group network and avoid loops. However, some social networks do not allow robots to crawl their network. For instance, Facebook bans robot accounts for a week. To overcome this issue, we used many users accounts. Our robot is able to change IP adresses, simulate human behaviour, switch between accounts, manage connection loss and save data in XML format and SQL database to avoid loops and replay attacks offline.

5 Link disclosure attacks

In this section we exploit the group uncovering attack detailed in previous section to perform link disclosure attacks. We aim to disclose two types of link: friendship between users and membership between users and groups.

5.1 Friendship and membership attacks

The attacker can explore the group networks of his target then check the member lists of distant groups to disclose group membership links to the masked groups. However, results show that less than 0.1 group membership in average can be disclosed by this attack. This can be explained by the fact that 78.84% of attacked profiles do not change group visibility parameters and even publish their memberships to secret groups. On the other hand, by exploring groups networks of 14,517 profiles we disclosed 430 different secret groups and 756 of their members. Secret groups can help to disclose communities if their member lists are disclosed. Moreover, their members can be taken into consideration to compute the probability of friendship between two users who hide their friend lists.

In this work we aim to disclose friendship links with certainty. In undirected social networks it is sufficient but not necessary that one of the two friends publishes his friend list to disclose the friendship link between them with certainty. In this perspective, an attacker can query all friend lists of the distant groups members of the target and check if he is listed in public ones. Opportunely, some social networks afford features that can be used to rapidly check friendships between users. For instance, friendship between two users of Facebook can be easily checked through the following PHP request (7):

$$/friendship/ < nid_1 > / < nid_2 > \tag{7}$$

Where $< nid_1 >$ and $< nid_2 >$ are numeric IDs of two different users. In fact, the request (7) returns the date of the link creation between two users if and only if there is a friendship link between them and at least one of them publishes his friend list. Taking advantage of this feature, attacker can easily

follow Algorithm 2 to disclose both friendship and group membership links of his target. Algorithm inputs are the profile of the target, the number of hops and the minimum number of links to disclose.

```
Data: t: target profile, h: number of hops, th: disclosed link threshold
Result: d_f: set of disclosed friends, di_g: set of disclosed groups
seedGroups ← getSeedGroups(t);
sizeSort(seedGroups);                          ▷ list of set of groups sorted by size
while d_f.size() < th  &  seedGroups.length()>0 do
    d_m2.addAll(d_m);                          ▷ d_m2 contains all tested profiles
    d_g.clear(); d_m.clear();
    explore(seedGroups.pop(), h, d_g, d_m);                    ▷ see algorithm 1
    d_m.removeAll(d_m2);                       ▷ remove already tested profiles
    for each m ∈ d_m do
        if friendship(m, t) then
            d_f.add(m);
        end
    end
                                   ▷ all newly explored groups are not tested yet
    for each g ∈ d_g do
        if getMembers(g).contains(t) then
            di_g.add(g);
        end
    end
end
```

Algorithm 2: Friendships and group membership attacks based on k-hop group graph traversal

We have attacked more than 100 active Facebook profiles that hide their friend lists from each set detailed in Section 3. For each attack we only checked the groups belonging to the same set to disclose friendship links between the target and those groups members. Note that users can be members of many groups from the same set. Since tiny groups densities are higher than large ones, fewer requests are required to disclose friendship links with certainty between the former members than between the latter members. 1-hop attack results (Figure 9 (a), blue curve) show that the average number of required requests to disclose one link with certainty increases as the size of groups increases. Only 6 requests in average are sufficient to disclose a friendship link with certainty of a target joining groups gathering less than 40 members against more than 7 requests in average for larger groups. However, the average number of requests to disclose one friendship link decreases if attacks involve 2-hop distant groups from the target. This does not mean that the ratio of published friendship links (PFLs), between the target and 2-hop distant groups members from him, is higher than

the ratio of PFLs between the target and 1-hop distant groups members from him. But, the ratio of PFLs between the target and the union of both 1-hop and 2-hop distant groups members from him is higher than any of the two ratios.

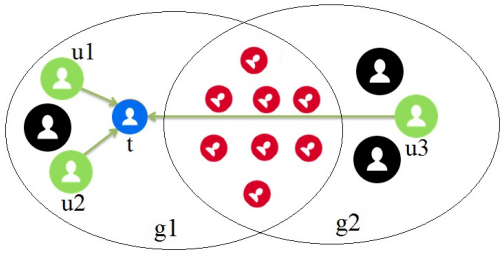

Fig. 6: 2-hop friendship disclosure attack.

Observations. Figure 6 gives an illustration of an observed social phenomena. Users within the same network tend to crowd in small and highly overlapping groups. Thereby, small networks pop up within big networks. To put it in another way, some members joining the same group (e.g., $g1$) decide to create a new group (e.g., $g2$) of similar size and to add some of their acquaintances to it. And so they act as gateways between both groups (inclined nodes in Figure 6). Some newly added members to the latter group (e.g., $u3$) publish their friendship links to the former group members. Therefore, the ratio of published friendship links between the target t and all members of the two merged groups (e.g., 3/14 for $g1 \cup g2$) is greater than the ratio of published friendship links between him and any of the two groups taken alone (e.g., 2/11 for $g1$ and 1/11 for $g2$). And consequently, the average number of requests to disclose one friendship link decreases as well as the number of disclosed links increases.

However, Figure 9 (a) shows that 3-hop attacks are less effective than 2-hop attacks. This result can be explained by the fact that the ratio of members publishing their friendship to the target among 3-hop distant groups is low. On the one hand, crawling those groups may orient the attack toward adjacent networks and dramatically increase the number of requests to disclose one link in average. On the other hand, it may disclose masked groups of the target. With this in mind, attackers can perform 3-hop or above attacks to only disclose masked groups of the target by checking public member lists then perform 2-hop attacks to disclose friendship links. Moreover, they can reduce the size of attacked groups after each hop to avoid crawling adjacent networks. Thus, they can effectively uncover the group network of the target and minimize the number of requests to disclose friendship links.

5.2 Mutual-friend attacks

The term 'mutual friends' stands for friends in common between two users. Mutual-friend attacks are performed between the target who hides his friend list

and another user to disclose a list of friends in common between them. In this section we exploit group uncovering attacks to perform mutual-friend attacks [6] between two members of the same network. Attacker can take advantage of the features afforded by social networks in order to list public mutual friends of two users. For instance, mutual friends of two Facebook users can be rapidly listed through the following Facebook request (8):

$$/browse/mutual\_friends/?uid =< nid_1 > \&node =< nid_2 > \quad (8)$$

Where $< nid_1 >$ and $< nid_2 >$ are the numeric IDs of two different users. Thus, the attacker can follow Algorithm 2 steps while replacing lines from 8 to 17 by the function described by Algorithm 3 to disclose mutual-friend links between his target and other users. Similarly to Algorithm 2, this algorithm inputs are the target profile, the number of hops and the minimum number of links to disclose. But it discloses mutual friends between the target and the groupe members rather than friendships between them.

```
1 for each m ∈ d_m do
2 |  d_f.addAll(mutualFriends(m, t));
3 end
```

Algorithm 3: Mutual friend attack

In fact, a mutual-friend request (8) between two users returns the list of their mutual friends that publish their friend list if and only if at least one of the two given users publishes his friend list as well. Starting from the hypothesis that a mutual-friend attack is performed between the target who hides his friend list and another user, it is only successful if both the latter and the mutual friend publish their friend list. Moreover, it is not effective in the case of sparse networks since it does not disclose friendship link between two users that do not have mutual friends even if one of them publishes his friend list. The example depicted by Figure 7 shows that despite the fact that User $u1$ publishes his friend list, mutual-friend requests cannot disclose the friendship link between him and the target t. Dotted arrows represent masked links and solid ones represent

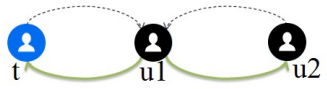

Fig. 7: Undisclosed links by mutual-friend attack.

public links. In this example only User $u1$ publishes his friend list and both User $u2$ and the target t hide theirs. Hence, the results of all possible mutual-friend requests between Users $u1$, $u2$ and t are empty since two of them hide

their friend list. However, friendship requests can disclose the friendship links between the target t and User $u1$ and between Users $u1$ and $u2$. Figure 8 depicts the average number of undisclosed links by a mutual-friend attack but disclosed by a friendship attack. We notice that this number increases with the number of hops.

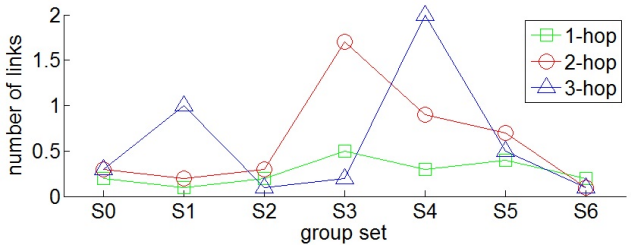

Fig. 8: The average number of undisclosed links by mutual-friend attack but disclosed by friendship attack.

Having said that, mutual-friend attacks can disclose more friends than friendship attacks if the target shares many mutual friends with his distant members. Figure 9 (b) shows that the number of mutual-friend requests to disclose one friendship link is quite similar for 1-hop and 2-hop attacks and increases for 3-hop attacks. However, it is far lower than the number of friendship requests depicted by Figure 9 (a) as mutual-friend request returns a list of friends.

To get better results the attacker can combine both attacks. For instance to maximize the number of disclosed links, he can sequentially perform a friendship attack after a mutual-friend attack. Hence, the number of attack requests will be equal to $2n - d$ where n is the number of distant groups members and d is the number of disclosed links between the target and them by mutual-friend attacks. Besides, he can alternatively perform both attacks to disclose friendship links between the target and his distant groups members. He can then follow Algorithm 2 steps while replacing lines from 8 to 17 by Algorithm 4 in order to focus his attack on distant groups members. Thus, the number of attack requests will belong to interval $[2; 2n]$. In fact, if mutual-friend requests do not disclose any friendship links between the target and his distant groups members then the number of attack requests will be equal to $2n$, by adding n friendship requests and n mutual-friend requests. On the other hand, if the target network is highly connected and the first mutual-friend request between the target and one of his distant groups members returns the rest of distant groups members then the number of attack requests will be 2, namely one friendship request and only one mutual-friend request.

```
for each m ∈ d_m do
    if (!d_f.contains(m)) then
        if (friendship(m, t)) then
            d_f.add(m);
        end
    end
    d_f.addAll(mutualFriends(m, t));
    if (d_f.containsAll(d_m)) then
        break;
    end
end
```

Algorithm 4: Mutual friend and friendship attacks

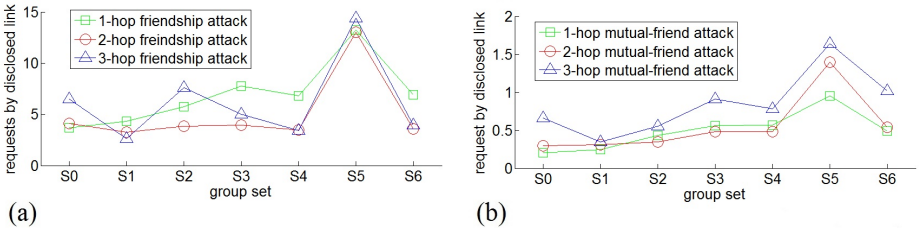

Fig. 9: Results of attacks: (a) The average number of friendship request to disclose one friendship link, (b) The average number of mutual-friend request to disclose one friendship link

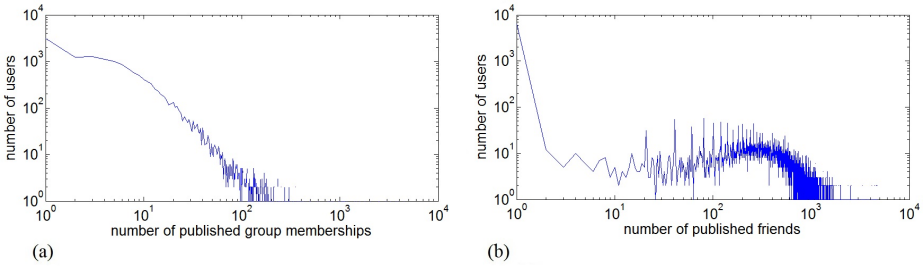

Fig. 10: Sample of 14,517 facebook profiles: (a) Frequency of published group membership, (b) Frequency of list of friends size.

6 Dataset

We have performed online attacks on Facebook. We have crawled 14,517 profiles, 22,855 groups and 76,772 mutual-friend lists. The resulting graph contains 4,153,379 user nodes, 131,410 group nodes, 5,720,973 friendship links and 1,225,533 group membership links. We noticed that 78.84 % of crawled profiles do not mask their groups 56.95 % publish their friend lists and 47.77 % publish

both. Among users who publish their friend list, the number of friends for a user in average is 530. And among all crawled profiles the number of unmasked groups for a user in average is 14.17. Figure 10 depicts the frequencies of published groups per user (a) and number of friends (b).

7 Conclusion

Friendship links on social networks hold sensitive information about the community structure and affinity between users. Disclosing them can expose users to the highest danger of leaking personal sensitive information such as political orientation. In this paper we have tackled the problem of link disclosure with certainty. We have performed online attacks on active Facebook profiles and proved that attackers can easily and rapidly disclose many hidden links with certainty taking advantage of social network APIs.

References

1. R. Bakhshandeh, M. Samadi, Z. Azimifar, and J. Schaeffer. Degrees of separation in social networks. In *Proceedings of the Fourth Annual Symposium on Combinatorial Search, SOCS 2011, Castell de Cardona, Barcelona, Spain, July 15.16, 2011*, 2011.
2. N. Barbieri, F. Bonchi, and G. Manco. Who to follow and why: link prediction with explanations. In *The 20th ACM SIGKDD, New York, USA - August 24 - 27*, pages 1266–1275, 2014.
3. R. Y. Dougnon, P. Fournier-Viger, and R. Nkambou. Inferring user profiles in online social networks using a partial social graph. In *28th Canadian Conference on Artificial Intelligence, Halifax, Canada, June 2-5*, pages 84–99, 2015.
4. S. Edunov, C. Diuk, I. O. Filiz, S. Bhagat, and M. Burke. Three and a half degrees of separation. In *Research at Facebook*, 2016.
5. I. Elkabani and R. A. A. Khachfeh. Homophily-based link prediction in the facebook online social network: A rough sets approach. *J. Intelligent Systems*, 24(4):491–503, 2015.
6. L. Jin, J. B. D. Joshi, and M. Anwar. Mutual-friend based attacks in social network systems. *Computers & Security*, 37:15–30, 2013.
7. N. Memon and R. Alhajj. Social networks: A powerful model for serving a wide range of domains. In *From Sociology to Computing in Social Networks - Theory, Foundations and Applications*, pages 1–9. Springer, 2010.
8. J. Scott. *Social Network Analysis*. Third edition. SAGE Publications, 2013.
9. J. Ugander, B. Karrer, L. Backstrom, and C. Marlow. The anatomy of the facebook social graph. *CoRR*, abs/1111.4503, 2011.
10. E. Zheleva and L. Getoor. To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles. In *Proceedings of the 18th WWW 2009, Madrid, Spain*, pages 531–540, 2009.
11. E. Zheleva, E. Terzi, and L. Getoor. *Privacy in Social Networks*. Synthesis Lectures on Data Mining and Knowledge Discovery. Morgan & Claypool Publishers, 2012.