OSN: when multiple autonomous users disclose another individual’s information

Cristina Pérez-Solà
Dept. d’Enginyeria de la Informació i les Comunicacions
Universitat Autònoma de Barcelona
Email: cperez@deic.uab.cat

Jordi Herrera-Joancomartí
Dept. d’Enginyeria de la Informació i les Comunicacions
Universitat Autònoma de Barcelona
Internet Interdisciplinary Institute (IN3)
Universitat Oberta de Catalunya
Email: jherrera@deic.uab.cat

Abstract—Online Social Networks (OSNs) are becoming more important in the web 2.0 paradigm. Although most implementations of OSN are not distributed applications, users conforming an OSN work autonomously posting their information in the OSN and interacting among them. Users are responsible of the information they post in their profile and, in the vast majority of social networks, they can limit the disclosure degree of such data regarding the members of the social network. However, a part from data they provide, other related data can be obtained from users: the relations between them. By appropriately crawling the web, it is possible to obtain information from a single user without accessing to his profile. In this paper, we present an attack to users’ privacy using a specific crawling algorithm that takes advantage of the network properties of OSNs.

Keywords: privacy, social networks, web crawling.

I. INTRODUCTION

The increasingly popularity of online social networks (OSNs) has lead them to become an important part of people’s everyday communication. With millions of individuals who use OSNs to share all kinds of contents, privacy concerns of how all this content is managed have arisen. Content shared in an OSN varies from trivial text messages to compromising photographs but, in either of those cases, users expect to control their shared data with their profile’s visibility configuration. In addition to this personal data, users in OSNs create relationships which can be also considered sensitive data from themselves. Moreover, the discovering of these relationships can also produce other data revelation, what makes link privacy an important issue to preserve in social networks. The disclosure degree of user’s relationships can usually be configured by the user. However, due to the bidirectional nature of this links in some OSNs, user’s privacy expectations may not be accomplished.

In this paper, we present an attack on users’ privacy based on the collected information from other users of the OSN. The attack exploits the specific characteristics of the OSN, regarding they social structure and aggregation.

The paper is organized as follows. Section II presents the state of the art. In Section III we provide some basic background on OSNs and social graphs. Section IV details the proposed attack based on a specific scheduler for a web crawler. Experimental results of the proposed attack are presented in Section V. Finally, section VI concludes the paper and provides some guidelines for further research.

II. STATE OF THE ART

Privacy implications of social networks have been a popular topic in recent years. In addition to personal data associated with each user, social networks include information on users’ relationships, which suppose an added risk to users’ privacy. Link privacy has been studied in [1] and [2]. In [1], Backstrom et al present several active attacks on edge privacy. These attacks allow an adversary to re-identify a set of targeted users from a single anonymized copy of the network. Active attacks require to modify the network before it is released, making them difficult to success on a large scale. They also present a passive attack in which users of the system try to find themselves in the released anonymized network, compromising the privacy of their neighbors. In [2], edge privacy is studied from the point of view of the number of compromised accounts needed to expose as much nodes as possible depending on the lookahead of the network. Lookahead is defined as the distance from which a user can see his friends links.

Theoretical work centered on maintaining privacy when releasing network data sets has also been done. In [3], the authors quantify privacy risks associated with different network release scenarios and propose a novel anonymization technique that leads to substantial reduction of the privacy threat while preserving the ability to estimate network measures with little bias. Although their technique reduces the privacy threat, it demands extremely strong structural requirements to the network in order to be applied, what makes it difficult, if not impossible, to use it in real world online social networks. In [4], the authors propose several strategies for preventing link re-identification in anonymized graphs. This strategies combined with node data anonymization achieve different levels of privacy preservation. However, their model does not take into account auxiliary information that an attacker may have obtained. Given the degree of data
disclosure in today’s online social networks, this auxiliary information can be obtained too easily to be omitted. In [5], authors assume that an adversary knows the neighborhood of some target individuals (what the neighbors of the victim are and how are they connected) and present an anonymization algorithm. Experiments show that many modifications have to be done by their algorithm to anonymize a network with average degree 4, which is far away from actual online social networks degree. In [6], other anonymization techniques are proposed, now considering that the adversary knows the degree of certain nodes a priori.

Much effort has also been done in re-identification algorithms for anonymized social graphs by Narayanan and Shmatikov in [7], where they present a deanonymization algorithm based on the usage of publicly available auxiliary information. Their demonstration with Netflix dataset proves that large scale deanonymization in real world networks is achievable with auxiliary information obtained from the public domain.

Although the Web crawling problem has also been widely studied, not much work has been done focused on crawling online social networks.

In [8], authors evaluate how node selection algorithms (among other parameters like the graph itself, the choice of seeds, the crawling size or the number of protected users) affect crawling efficiency and introduce biases in graph measures. These biases can be avoided by selecting the proper scheduler algorithm as is shown in [9], where a random sample of Facebook users is collected using a Metropolis-Hasting random walk. They also demonstrate that metrics obtained with MHRW largely differ from those obtained with BFS. Graphs retrieved by different collection techniques are also compared for Twitter network in [10].

Improving crawling performance gains importance as OSNs grow. Some work has been done in this direction by defining architectures for parallel OSN crawlers in [11].

Large-scale measurement of online social networks has been done in [12], where it is confirmed that online social networks satisfy the power-law, small-world and scale-free properties.

Although some of this crawling literature refers to users’ privacy, it always does it in terms of how users’ profile visibility affects the crawling process or to study how users tend to configure their profile’s visibility. Comparisons made by different crawling algorithms are centered in their effect on classic graph metrics or on crawling efficiency, but, to our best knowledge, no work about how crawling algorithms affect user’s privacy can be found in this crawling literature. A similar situation is found in the privacy literature, where user’s privacy is analyzed in detail but no references to crawler algorithms can be found.

III. ON-LINE SOCIAL NETWORKS

A social network is a social structure made by individuals which are tied by some kind of interdependency or relation. Online social networks (OSN) are web services that allow users to:
1) create a public (or partially public) profile,
2) establish relationships with other users,
3) share information about these relationships.

In general terms, joining an OSN consists of two basic steps. In the first place, users sign up by filling an online form with personal data which will conform the user’s profile. The visibility of this profile depends mostly on the OSN and, in second term, on users’ preferences. While some OSN as Tribe1 or Friendster2 make profiles public by default and allow them to be indexed by search engines, other networks such as Facebook3 or Flickr4 let users to configure their profile’s visibility based on groups. Other OSNs like Last.fm5 allow users to change the visibility of some of the attributes that the network stores while other OSNs like Twitter6 do not provide such granularity and the profile only admit two protection levels: visible or private.

Once the profile has been created, users can start to establish relationships with other users. There are many kinds of relations that a user can create in an online social network. “Friend”, “fan”, “contact” or “follower” are the most popular ones. Relations in an OSN can be unidirectional (usually requiring only one user approval in order to create them) or bidirectional (requiring consensus of both users to be established). Depending on the type of relationship described, using unidirectional or bidirectional links will make more or less sense. Some OSNs like Facebook are based on mutual approved bidirectional relations representing friendship whereas other OSN like Twitter are based on unidirectional relations which represent reading interests.

A part from creating a profile and establishing relationships with other users, OSNs usually offer other services like instant messaging, microblogging, online games or photography sharing.

A. Social graphs

Social networks are usually visualized as graphs, where nodes represent individuals (or organizations) and edges describe interaction relationships among them. In a social graph, edges can express different kinds of relationships: economic, friendship, sentimental, academic collaboration... Depending on the kind of relationship expressed, social

1http://www.tribe.net
2http://www.friendster.com
3http://www.facebook.com
4http://www.flickr.com
5http://www.last.fm
6http://twitter.com
Social graphs can be seen as directed or undirected graphs. Expressing social networks as graphs allows to use a whole variety of algorithms and metrics from graph theory to analyze and describe the network.

Social graphs have some specific characteristics that distinguish them from other graphs. One of the most outstanding one is the distribution of degrees in a power law [12] such that the probability that a node has degree \( k \) is proportional to \( k^{-\alpha} \) for some \( \alpha > 1 \). \( \alpha \) is called the power law exponent. Therefore, social graphs use to have a few nodes with very high degree and a lot of nodes with small degrees.

Social graphs are also small-world networks [12]. In a small-world network almost any node can be reached from every other node by a small number of hops and such networks present a community structure, with nodes highly connected within the same community and poorly connected between different communities.

### B. Social graph information

There are many metrics that provide information about a social graph. For instance, diameter and radius offer information of distances inside the graph and density is used to measure the tightness of the graph. Centrality metrics are used to measure how important, influent or powerful is a node based on its position inside the network. Some other metrics are related to the substructures that are found in the network. Clustering coefficient is an example of this last group.

Clustering coefficient is a measure of how well connected the neighborhood of a node is. When the neighborhood of a node is fully connected, the clustering coefficient is 1 whereas when there are no connections between one node’s neighbors the clustering coefficient is 0. More precisely, clustering coefficient of a node \( v \) is defined as the number of connections between \( v \)'s neighbors divided by the number of possible connections that could exist between them (for a undirected graph with \( \deg(v) = n \), that is \( \frac{n\times(n-1)}{2} \)). The local clustering coefficient of a graph is defined as the mean of all nodes’ clustering coefficients.

Social networks are small world networks which are known to exhibit high clustering coefficients. For this reason, nodes in social graphs tend to group together in small groups within the network connections are dense but between which they are sparser. High clustering is easily understood when speaking about social connections. Take as an example two adjacent nodes \( v \) and \( u \) (hence users \( v \) and \( u \) are, for instance, friends), and a third node \( y \) which is also adjacent to \( v \). It is more likely that \( y \) will also be a friend of \( u \) than another randomly selected node of the graph.

When the clustering coefficient of a node \( v \) is 1 all nodes in \( v \)'s neighborhood are fully connected or, in other words, form a complete subgraph. In network analysis terminology, complete subgraphs are known as cliques and are a highly studied phenomenon. Nodes that belong to the same cliques in social graphs usually share some common properties like age, ideology, beliefs, behavior or food habits [13]. However, the use of cliques in social graph analysis has been criticized for its overly restrictive nature. This has yield to other less restrictive subgroups definitions like \( N \)-clique, \( N \)-clans, \( K \)-plexes or \( K \)-kernels.

There exists a large set of algorithms for discovering community structure in networks which take different approaches to the problem. Some of them are based on cliques. Other methods define similarity measures between nodes in the network and then group together nodes which are more similar. In contrast, there are algorithms which start from the whole graph and iteratively remove edges which high betweenness, getting communities to become disconnected components of the graph.

Given a graph, there exists many possible communities divisions which meet the requirement of having nodes highly connected within the community and poorly connected between different communities. Dendrograms are used in order to represent these different levels of communities divisions.

### IV. Proposed attack

As we have seen, social graphs present high clustering values and users tend to form cliques and other highly connected structures. Such properties may be exploited to obtain information of a specific user from the information their friends share, without accessing the victim’s profile.

#### A. Attack scenario

We consider an online social network which allows its users to configure their profile visibility as either totally private or totally public. User’s profile include personal data (which will be considered node attributes) and user’s relationships (edges). Our victim (\( u_0 \)) is a member of this OSN and has his profile configured as totally private and, therefore, no one can see his personal information nor his relationships with other users. However, \( u_0 \) has \( n \) relationships with other users that have configured their profile as public and, therefore, everybody can take a look on their personal data and their relationships.

Given this scenario, the attacker goal is to obtain information from the victim (\( u_0 \)) without accessing to his profile. We assume that the attacker already knows \( r \) friends of \( u_0 \), for some value \( 1 < r < n \).

#### B. Retrieved information

As we have already mention, information about a user of a social graph is essentially of two different types: node attributes (personal information about the user which can be found in his profile) and structural information (which includes node relationships).

Our attack is designed to obtain structural information about \( u_0 \) through \( u_0 \)'s friends. Such structural information includes \( n \) degree.
It is worth mention that, although the attack does not provide specific node attributes (since our model assumes $u_0$ has configured his profile as totally private), node attributes (i.e. personal information) can also be induced by discovering $u_0$’s friends. As we have seen, social networks are structured in communities and users in the same community are known to share some attributes. Discovering this communities and obtaining personal information on other users’ profiles can lead to discover this attributes from $u_0$. Moreover, some kind of personal information shared by users in social networks, like photography or videos, can be a direct source of other user’s personal data.

C. Attack description

We execute the attack through a specifically designed web crawler. Web crawlers are programs that automatically explore web pages in a methodical manner. Web crawlers start the search in one or more URLs, which are called seeds, and explore them in order to find new URLs to search for, until they reach a predefined termination condition. When used to crawl OSNs, web-crawlers start from an initial user, or list of users, and discover other users of the network by following their social relationships. Web crawlers distinguish two different types of user retrieval: discovered and crawled. A user is discovered when it has been detected by the crawler and user is crawled when all his friends have been discovered.

The order in which this relationships are followed is determined by the scheduler algorithm. Such algorithm is extremely important and determines the quality of data the crawler will obtain.

The most popular scheduler algorithm is Breath First Search (BFS), where the next node to crawl is simply selected by picking up the first node in the queue. Newly discovered nodes are appended at the end of the queue, thus first nodes discovered are the first ones to be explored. For that reason, BFS is also known as First In First Out (FIFO).

A straight forward method to obtain $u_0$’s information would be to set the victim as the initial seed of a crawler with BFS scheduler algorithm. However, the assumptions of our attack (see Section IV-A) impose that $u_0$ has defined his profile as totally private, so the crawler could not access to $u_0$.

For that reason, we have designed a scheduler algorithm that simulates what a BFS centered on the victim would do. Starting with one of the $u_0$’s friends, the new algorithm, called outliner, tries to crawl $u_0$’s neighborhood without exploring the node $u_0$ itself. The algorithm that we propose takes advantage of the high clustering coefficient showed by social networks, that is translated in a high probability that a friend of $u_0$ and $u_0$ itself share more than one friend.

We present a specific crawling scheduler algorithm to maximize the amount of information acquired from $u_0$ without explicitly crawling his profile.

1) Scheduler algorithm: The outliner algorithm maintains a list of waiting-to-explore nodes with their known distance to the victim. At the beginning of the crawl, the list is initialized with the victim’s friends (that is, $r$ nodes adjacent to the victim’s node, known by the attacker, with $1 < r < n$) set at distance 1. Every time a node is crawled, all of his friends are added to the waiting-to-explore list with distance incremented by one, obviously discarding nodes already discovered. This distance can be inaccurate because it is based only in the attacker’s partial knowledge of the network. When a node is crawled, it is possible that a new relationship with the victim’s node is found. For this reason, every time such relationship is found, it is needed to recalculate node distances with the new information found. Such recalculation allows to take posterior scheduling decisions with as much information as available. Once a node has been crawled, the new node to crawl is the one with the lowest distance value from the list of waiting-to-explore nodes. This assures that the crawler will remain as close as possible to the victim.

V. Experimental results

In this section, we present a proof of concept of the proposed attack by implementing it over two different online social networks. Results provided give a flavor of the attack performance on different OSN with different structure.

A. Experimental set-up

The proposed attack has been implemented in two different OSNs: flickr and last.fm.

Flickr is an online photography sharing community. It is used by bloggers and webmasters to store images that will be embedded in web pages as well as by photographers who share and comment their creations. They claim to have more than 4 billions of images in their system and 32 millions of registered users.

Last.fm is a music recommendation system and an Internet radio streaming service. It builds the user’s profile by analyzing his musical preferences based on the music he listens on last.fm radio stations. Last.fm system is also capable of analyzing music that the user listens on his own music player via some specific plugins. Last.fm network is made by more than 30 millions of active users.

We have chosen this two OSN because network data closely related to the victim present different degree and clustering values (see Table I). Such differences allow us to study how OSN structure affects the performance of the proposed attack.

We target a user $u_0$ in each social network as a victim. In order to test our attack in a worst case scenario, we restrict the knowledge of the attacker to only one friend of $u_0$ (that is $r = 1$).

As we detailed in Section IV-C, the attack is performed through a web crawler which crawls the OSN looking for
Table I
VALUES FROM CRAWLED GRAPHS OBTAINED WITH BFS (SEED $u_0$).

<table>
<thead>
<tr>
<th>OSN</th>
<th>Average clustering coef.</th>
<th>$u_0$ clustering coef.</th>
<th>Crawled mean degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lastfm</td>
<td>0.167</td>
<td>0.031</td>
<td>80.6</td>
</tr>
<tr>
<td>Flickr</td>
<td>0.343</td>
<td>0.103</td>
<td>336.6</td>
</tr>
</tbody>
</table>

the desired information. The termination condition of the crawler is related to the final goal of the attack. A final goal of such attack could be the obtainment of all friends of the victim. However, if the cluster coefficient of the nodes involved in the crawling is low, the attack will take too much time to finish. Furthermore, the degree of the crawled nodes also affects the attack’s performance. In order to test the performance of our attack regarding different properties of the OSN, we have fixed the goal of the attack to obtain more than one third of the total friends of the victims, which we consider that is a reasonable amount.¹

B. Data analysis

Figure 1 shows the data obtained from the attack in last.fm. The Figure shows the complete 1-node neighborhood graph centered on the victim, where solid lines are the ones obtained by the attack while dotted lines are existing relations that the attack has not retrieved. In this scenario, the victim $u_0$ (the center node) has degree 27, then the attack finishes when 10 nodes are obtained. The seed of the crawling algorithm for the attack is the node denoted with a circle.

Figure 2 shows the data corresponding to the Flickr network with the same notation as in Figure 1. In this case, the victim $u_0$ (the center node) has degree 35, then the attack finishes when 12 nodes are obtained.

Both attacks rise the objective of the 1/3 bound of the victim’s friends. Notice that, in the last.fm case, the crawler is able to connect three apparently disjoint subgraphs (central node is not crawled) since this disjunction is only at one hop level. Regarding the flickr case, Figure 2 shows that the number of nodes that can be discovered by this crawling algorithm is potentially bigger since the crawler still could discover existing relation between discovered (but not crawled) nodes and the victim.

A special consideration has to be done regarding the seed of the crawling algorithm. Notice that we have assumed that the attacker knows one of the victim’s friends ($r = 1$). However, the exact friend selection is important for the crawling algorithm. In case the selected friend is in a non-connected subgraph, the attack could not reach the objective. However, the probability of such event is related to the

¹We are aware that such assumption implies the knowledge of the degree of the victim, but we use such information to test the performance of the attack. Other termination conditions not related with that value could be defined.
number of nodes the attack can obtain with the best possible seed. Notice that, in our scenario, assuming that the friend the attacker knows is a random one, the probability to achieve the 1/3 bound at this neighborhood level is exactly 1/3, since this is the number of nodes that are in the same connected component.

Finally, it is worth mention how clustering coefficient and node degree of the OSN affect the performance of the attack. On one hand, the number of crawled nodes depends on the clustering coefficient. In the last.fm case, the crawler needs to crawl 805 nodes to achieve the objective (the 1/3 bound), while the number of nodes crawled in the flickr network is much lower, 475, for the same objective. This data is consistent with the fact that the clustering coefficient of flickr is greater than the one of last.fm (see Table I).

On the other hand, the node degree of the OSN determines the number of nodes the crawler has to discover. In our tests, the last.fm attack needs to discover a total of 69406 nodes to determine the friends of the victim while in the flickr case this number rises up to 124590 nodes. Again, this data is consistent with the mean degree value shown in Table I.

VI. CONCLUSIONS

In this paper, we have presented a privacy attack to online social networks (OSNs) users. The attack is performed through a specifically designed web crawler algorithm that exploits the inherent network structure of the OSN. We have presented experimental results showing that the attack is able to recover a relevant percentage of the relationships of a targeted victim without exploring his profile. We have shown that, even exploring only a small part of the network, it is possible to compromise a considerable amount of the targeted user’s neighborhood. This kind of attacks are further evidence that preserving privacy in network data is much more complex than in relational data.

Future research is focused on determining the adequate termination condition for the crawler based only on the attacker’s knowledge of the network. Analyzing in detail the effect of the clustering coefficient and mean degree on the success of the attack and the time needed to conduct it is also work that remains to be completed.

ACKNOWLEDGEMENTS

This work was partially supported by the Spanish MCYT and the FEDER funds under grant TSI2007-65406-C03-03 E-AEGIS, CONSOLIDER CSD2007-00004 "ARES", funded by the Spanish Ministry of Science and Education and MyCity project TSI-020501-2008-191.

REFERENCES


