On Anonymizing Social Network Graphs

Anne V.D.M. Kayem, Azhar Deshai, and Stuart Hammer
Department of Computer Science
University of Cape Town
Private Bag X3, Rondebosch, Cape Town, 7701
Email: akayem@cs.uct.ac.za, dsxazh001@myuct.ac.za, hamstuh89@gmail.com

***Abstract*—The proliferation of social networks as a means of seamless communication between multiple parties across vast geographical distances has driven an increased interest from government organizations and companies. Government organizations typically seek access to these pools of personal data for statistical purposes while companies tend to look at this data from a marketing perspective. Users typically post information containing personal data during social network interactions with other users because the aim is to share this information only with persons that are authorized to access the information. However, the growing desire to exploit this information for statistical and marketing purposes, for instance, raises the question of privacy. It is therefore increasingly important to come up with ways of anonymizing personal data in order to circumvent privacy violations. Previous work has focused on two major approaches to anonymizing data namely, clustering and graph modifications. Both techniques aim to preserve the utility of the data for analysis and keep the identities of the users secret. We postulate however, that both approaches are in fact vulnerable to privacy violations and so do not enforce the property of anonymity. In addition, we argue that this problem is in fact NP-Hard and that the difficulty is in identifying as well as anonymizing all the possible channels that might leak information about a person's true identity.**

Keywords - Anonymization, Privacy, Social Networks, Graphs

I. Introduction

The Internet abounds with web-based Social Networking platforms that range from those that are geared towards friendships to the ones that target academic and business circles. The popularity of these platforms is due, in part, to the fact that web-based social networks provide an ideal platform where multiple users can interact seamlessly and share information flexibly. In this way, web-based social networks have benefited from the development of Web 2.0 applications [2] that facilitate information sharing, interoperability, user-centred design, and collaboration on the World Wide Web [6], [22]. Examples of friendship social networks include FaceBook [5] and Twitter [17], while examples of academic and business social networks include Mendeley [11] and LinkedIn [10].

In general, web-based social networks aim to model social relationships with the help of graph structures which consist of vertices and edges. For simplicity, we will hereafter refer to a ***"web-based social network graph"*** as a ***"social network"***. In a social network, a vertex or node represents an individual (social network user) and an edge represents the relationship. For instance, in a friendship social network, an edge would represent a friendship, between two or more individuals belonging to the social graph. These relationships and connections are often beneficial to third parties such as commercial enterprises and individual users [6], [13], [22]. Commercial enterprises use social networks to identify potential clients or even sometimes to recruit possible employees. While individual users use social networks typically to contact friends, have discussions, or just to stay in touch with acquaintances/friends. Consequently, social networks have become an integral part of the lives of a lot of people. In fact, the FaceBook [5] social network claimed in 2012 to have registered more than 901 million users as of March 2012 [5], [15].

By their very nature, social networks manipulate large volumes of personal information in an environment that is relatively open. Authentication and authorization procedures are typically implemented to enforce privacy in social networks ensuring that only users with valid permissions gain access to information. In addition to this, anonymization techniques are also used to "hide" demographic information in order to ensure that other sensitive bits of an individual user's information is not accessible even to the users belonging in his/her social network [13], [19]. For instance, in FaceBook [5], [16], [18], a user can choose to declare how much information he/she would like to share with the public or with his/her friends by defining levels of "trust" for different user groups.

Yet, in spite of the fact that security mechanisms

have been widely implemented on social networks, their very nature makes social networks inherently prone to unauthorised information exposure [19]. For instance, in July of 2010, a security consultant (Ron Bowles) used a program to collect personal data off Facebook [5] and publish it on Pirate Bay (a popular file sharing site) [12]. Social network security violations, such as this one, arise because of the difficulty of anonymizing and controlling the flow of information in these environments. In Figure 1 for instance, a user, say Alice, might choose to anonymize her personal data so that only certain portions are visible to close friends like Bob, and none at all to acquaintances like John. However, John might be able to infer bits of personal information about Alice, if he is in at least one of Bob's "trusted" friends groups and additionally, gains transitive access to information from Bob downloads. For instance, as shown in Figure 1 Bob could download photographs from Alice's profile and proceed to share them with John. In this way, John might guess that Bob is Alice's friend and can also make other deductions about Alice's lifestyle and personality, from information in the picture that is difficult and/or impossible to anonymize. Therefore, anonymizing social network data to enforce privacy is challenging because it is difficult to monitor all the possible ways in which the information may flow or be shared.

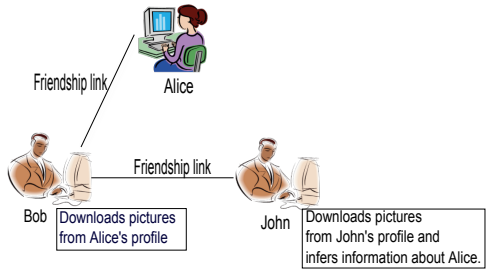

Fig. 1. A Simplified Information Sharing Scenario

In this paper we discuss two major techniques that have been used to anonymize social network graphs namely, clustering and graph modifications. The difficulty of addressing this problem rests on the fact that it is equivalent to the problem of identifying and eliminating all the possible channels that might leak information about a user's true identity. This is in fact equivalent to the problem of computing an independent set of the social network graph based on the anonymization criteria, to ensure that anonymized information is never leaked to unauthorized users. We postulate that both the clustering and graph modification techniques for anonymizing social networks are vulnerable to privacy violations because obtaining an optimal solution for the independent set problem is known to be NP-Hard.

The rest of the paper is structured as follows. In Section 2, we present related work on modelling anonymity and discuss the clustering as well the graph modification approaches. We build on this related work, in Section 3, to discuss three attacks on social network graphs that stem from the difficulty in eliminating channels that might leak information. In Section 4, we postulate that the challenge in anonymizing social network graphs stems from the fact that the problem is NP-Hard and so, computationally difficult to solve. Therefore, any solutions we get in these scenarios will generally be based on heuristics drawn from known attacks possibilities. Finally, in Section 5, we offer concluding remarks and some discussion of future work.

II. Related Work

Anonymity of information has been widely researched with respect to relational databases. In general, private data in relational databases can be re-identified by joining anonymous tables with some external tables that model the background knowledge of the attackers [6], [22]. In order to do this effectively, several anonymization techniques have been developed for keeping relational data secure. These methods include techniques like k-anonymity [20], l-diversity [8], and t-closeness [14]. Each of these techniques can only be effective on relational data sets and cannot easily be applied to social network graphs [6], [22]. The complexity in transferring these anonymization techniques to social network graphs emerges in the interrelations between the users and the data. These interrelations make anonymizing the social network graph challenging because a trade-off between usability and security is required to ensure that the users can access their data but that at the same time it is kept secure from unauthorised access.

In general approaches used to anonymize social network graphs can be put into two categories namely, clustering and graph modification approaches. Clustering based methods group a set of nodes and edges into a subgraph and then anonymize the subgraph by transforming it into a super-vertex [6], [22]. In the vertex clustering approach, the super-vertices are formed by grouping nodes (vertices) into partitions [6], [7], [22]. The number of nodes in each partition as well as the density of the edges within and across the partitions is published. By performing this sort of anonymization, the data can be aggregated into a single point where it is easily accessible but is stored in such a way that makes it hard for an attacker to accurately guess what the identities of the users are. In Figure 2(a.) the vertices are still representatives of individuals and their data, whereas in Figure 2(b.) vertex clustering

makes distinguishing individuals much harder. When

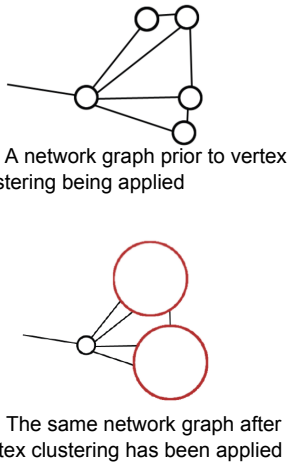

Fig. 2. Vertex Clustering

links between nodes represent flows of information it is more useful to anonymise the links between the nodes as opposed to anonymizing the nodes as shown in Figure 2(b.). For instance, in the FaceBook [5] social network, it might be more useful to anonymize the connections that link a user to set of friends as opposed to anonymizing the nodes. Edge clustering focuses on protecting edges that are sensitive with the idea being to anonymize the edges so that attackers cannot infer that the edges exist. One method is to remove all the edges in the graph but this reduces the usability of the anonymized graph [6], [22]. Zheleva and Getoor [21] proposed removing edges according to the probability of a sensitive edge existing between the nodes. They present three heuristic algorithms that protect edge anonymity using edge deletions. Experimental results based on three real-world social networks and several utility measures, show that these algorithms effectively preserve edge anonymity and acceptable anonymous graph utility. However, this approach only guarantees that the edges of some known sensitivity are known beforehand and can be anoynmized. Therefore when the sensitivity of an edges changes the algorithms need to be run again and the sensitivity thresholds reevaluated. Other approaches remove only the sensitive edges while maintaining all the other edges [21]. Reducing several nodes and edges into a single vertex results in a shrunken anonymized graph which hinders analyses of the local structure of the social network [21].

Graph modification techniques emerged as an extension to edge clustering approaches with the idea of inserting and deleting nodes as well as edges in the graph [6], [22]. Graph modification approaches modify the structure of the graph to preserve the scale and local structure of the graph. Techniques for modifying the graph include random and greedy graph modifications. Both techniques give the observer the impression that the graph has remained unchanged, while "hiding" sensitive information from unauthorised access [6], [22].

In random graph modifications a number of edges are inserted and deleted from various points in the graph at random in order to create the anonymized graph. The vertices in these graphs are not changed but rather the goal of this form of modification is to unsettle the network graph. Therefore if an attacker were to attempt to re-identify individuals in the social network graph, by using external information, he/she will not be able to simply exclude the vertices that do not match his/her background knowledge of the vertex. This is because the target vertex may have been modified, making it hard to correctly identify the vertex [22].

The greedy graph modification approach builds on the random graph modification technique by extracting knowledge about the neighbourhood of each of the nodes, considered to be the most vulnerable to network attacks, and greedily grouping them according to background similarity [6], [22]. However, as with the previous approaches, the random and greedy graph modification techniques rely on some apriori knowledge of sensitive information that is known when the graph is formed. If the changes occur after the formation of the graph and in ways that were not considered beforehand, the information involved in the change goes unanonymized. Consequently, previously anonymized information can get leaked through inference, to unauthorized users.

III. Attacks on Anonymity

As mentioned before, attacks on social networks are relatively frequent because of the large amount of personal information on these networks. When social network graphs are anonymized, attackers attempt to correlate or manipulate information in the graphs to expose users' sensitive information. In general attacks on anonymity are divided into two categories namely, active and passive attacks. Active attacks assume that the attacker is able to modify the network graph prior to its release by interacting with the users, while passive attacks use the anonymized graph structure to infer information about the users without interacting with the users. In this section we discuss three of the attacks that have been presented in the literature to highlight the challenges involved in anonymizing social networks.

A. Embedded Subgraph Attack

The embedded subgraph attack is an active attack that was developed by Backstrom et al. [1]. They created a set of attacks that allow attackers to identify whether relationships exist between users in an anonymized social network. These attacks rely on the fact that the attacker is able to modify the network prior to its release.

The attack is carried out by selecting a set of users whose privacy is to be violated and then creating a small number of new user accounts that are also known as Sybil nodes (vertices) that have relationships with the target users. A pattern of links is then created between the user accounts to make them stand out in the anonymized network graph.

As an example of how the embedded subgraph attack works, consider that a user say, Alice is an attacker and that she wishes to de-anonymize three target users (Tom, Dick, and Harry) in a social network like Facebook [5]. Alice begins by creating five new user accounts which she then adds to a frinedship group. This creates a regular pattern of links between these five users. Next, she attempts to form relationships with Tom, Dick, and Harry - her targets - by attempting to add them to her friendship group in order to embed a malicious subgraph in the social network. If Alice is successful, the released anonymized social network would contain the embedded subgraph and a quick search of the previous patterns would enable the re-identification of the target users.

However, active attacks present three primary limitations in their structure and in the way they are designed to occur. First, it is difficult to stage large-scale attacks on social networks because social networks like FaceBook [5] have implemented structures in the authentication mechanisms to prevent this from happening. FaceBook [5] for example, checks for uniqueness of users by verifying email addresses which makes it difficult for an attacker to create a large number of dummy nodes. However, a large number of dummy nodes can be created if this were a concerted criminal activity with several attackers involved in the process.

Second, the attacker has no control over incoming links to the Sybil nodes [9] and most users of social networks would not link to dummy nodes because the person who owns such a node is unknown to the user. Therefore the attacker will have created a node with many outgoing links but relatively few incoming links. This will make the node stand out in the network graph making it more likely to be detected by the security administrator [9]. Sybil attacks however, are quite challenging to detect because attackers usually try to construct Sybil graphs based on some stolen personal information that impersonates a legitimate user. For instance, high school records are not often kept very secure and so it is sometimes relatively easy to get such information and create a user profile. People also don't easily remember high school classmates and a person with hundreds of friends, might be more tempted to include an unknown user on his or her profile than one with less.

Third, in most social networks there needs to be mutual linking before information is made available in any form. Therefore, a genuine user has to first accept the dummy node's friend request before information is made available between them. Since the social networks are formed on the basis of user behaviour it is hard to predict how the social network graph will be constructed and how one might anonymize the links between the nodes effectively. We also note that even though active attacks are infeasible for large-scale attacks, they are useful in creating a starting point for implementing large-scale passive attacks.

B. Coalition Attack

The coalition attack is an example of a passive attack that was developed by Backstrom et al. [1]. This attack relies on the concept that social network users typically belong to a small and uniquely identifiable subgraph [6], [22]. A set of users collude to form a "fictional" subgraph, by using their knowledge of the network composition, to infer details about other users in the anonymized social network. In order to explain how this happens, consider the case of Alice who wishes to de-anonymize a target set of individuals in a social network. Alice colludes with some of her friends to form an alliance in which the users all know the details about each other. In this way, the users belonging to the alliance can discover information about users outside the alliance, but who are linked to those within the alliance. So in spite of the anonymized social network, users belonging to Alice's alliance can proceed to de-anonymize their neighbours data. In contrast to the embedded subgraph example, no modification to the social network occurs. The attacker gains knowledge about a user by proximity, and inference. Basically by forming a seemingly legitimate network, Alice's alliance can entice other unsuspecting users to join the group thereby making the outside users more vulnerable to de-anonymization attacks. However, Narayanan and Shmatikov [13] point out that the coalition attack is not feasible for large-scale de-anonymization and the colluding users only undermine the privacy of users who are already their friends.

C. Auxiliary Information Attack

Narayanan and Shmatikov [13] devised an attack that combines some active and passive attack features. For simplicity, we term this attack the "auxiliary information attack". This attack is used for large scale re-identification of individuals and attack assumes that auxiliary information is made publicly available to the attacker. Auxiliary information is classified as any additional information or attributes that an attacker knows [13]. The primary form of auxiliary information used for this attack consists of another social network that has partially overlapping membership.

The attack requires two graphs, the anonymized network graph and the set of auxiliary graphs that overlap the anonymized graph. The attack then runs in two stages, first the attacker locates a small number of nodes that are present in both graphs and maps them to each other and second, the initial mapping is extended to other nodes using only the layout of the network. These new re-identifications are re-submitted to the auxiliary information attack algorithm until there is a large mapping between both graphs thereby de-anonymizing the information in the anonymous social network. A simple and high-level example of how this occurs is as follows.

Let us assume that in reality Alice (our attacker) is a legitimate friend of Bob's but that Bob has no idea that Alice is interested in gaining illegal access to the information of other users of the social network. Alice knows that Bob is friendly with another man at work and she (Alice) knows what Bob's friend looks like. The relationship between Bob and the other man and his appearance serves as Alice's auxiliary information. Alice wants to find out as much as she can about that man. So, she maps this information to a social network by searching through Bob's friends. She locates the man, who turns out to be Tom which adds to her auxiliary information. Alice can then search Tom's profile to find out more about him, and his friends'. Alice can keep adding the new information to her auxiliary information until she uncovers everything she can about Tom by mapping her auxiliary information to an anonymous target. This is a very high-level example to promote the understanding of the concept of this attack, but serves to show that inference techniques are quite effective in circumventing anonymization schemes in social networks.

The main limitation here is that the attacker needs the information of at least two subgraphs with in a social network to successfully deanonymize the information of a user who has no direct edge links to the attacker. However, in this next section we argue that by combining attack strategies, circumventing anonymization schemes is made easier and that the challenge in preventing these attacks from occurring rests on the fact that this problem is equivalent to the decision version of the independent set problem which is NP-hard. Therefore, there is no optimal polynomial time algorithm to completely prevent de-anonymization attacks from occuring and so proposed solutions, only mitigate attacks based on known de-anonymization possibilities.

IV. Why More than Anonynmized Graphs?

We postulate that the difficulty in preventing de-anonymization attacks from occurring is due to the fact that completely anonymizing a social network is in fact equivalent to computing an independent set of the graph. That is, to be successful, the anonymization technique needs to simulate the computation of an independent set in eliminating all the sensitive links that might lead to a discovery of personal information. This is a challenging problem because it requires first identifying all the possible channels through which such violations might occur, enumerating them, and classifying them according to some priority. Additionally, since the channels are identified according to known attack possibilities, the anonymization schemes are only as good as what is known about how the attacks might occur.

A. Re-identification Algorithm

Wondracek et al. [19], developed a de-anoymization approach that extends the re-identification algorithm [13] with modifications that are based on the attacker's knowledge of a user's group memberships. The attacker has access both to a target social network that has been anonymized, and to an auxiliary network where users overlap the anonymized social network. De-anonymization happens in two steps namely, node re-identification and propagation. Node re-identifications are typically used to extract sensitive personal information in anonymized social networks and can be comibned with auxiliary information to discover the exact identity of a user in an anonymized social network. The next stage is the propagation stage in which identified nodes and connections are used to identify more nodes in the target network. This process is repeated recursively, resulting in a mapping between large portions of the networks.

Many large public social networks have overlapping user profiles which is useful in running the re-identification algorithm. The de-anonymization algorithm in its unmodified form has been shown to be reasonably robust against certain graph modifications. This is partly because the unmodified de-anonymization

algorithm exploits the fact that the nodes that are re-identified provide more auxiliary data. However, in spite of the existence of the auxiliary data, re-identification is non-trivial because there needs to be some connection between the attacker and at least one member of the target group.

The re-identification attack is passive because it requires only access to an anonymized social network and an auxiliary one. Whereas, active attacks require inserting a large number of nodes into the graph before it is anonymised and released. Inserting a large number of nodes to anonymize a social network graph is often impractical and well defended against by social network operators, so the technique does not scale well.

Typically the attacker would know more than the two social networks discussed, that is the anonymized and auxiliary social network. Additionally, the attacker has detailed knowledge of a small number of nodes that appear in both networks. In general, the two social networks (the anonymized and auxiliary graph) have many members in common because in general Internet users belong to two or more social networks. Narayanan and Shmatikov [13] demonstrate that by using usernames and other contextual information, nodes in one of the social network (the anonymized or auxiliary networks) can be mapped to the other social network with reasonable accuracy. As well, the size of the social networks ensures that there is a strong chance of overlap.

Once the attacker has succeeded in forming a social network graph that he/she can de-anonymize, in the propagation stage, a seed algorithm is used to search for nodes that closely match the structure of the auxiliary network. The reason for doing this is that there is usually a close similarity between a connected sub graph in the auxiliary network and a similar graph in the target network. That is, if a group of users share some similar interests (e.g. friendship) and one can deduce this from leaked private information, then there is a good possibility that in the real social networks these connections will persist. The seed algorithm searches for nodes that closely match the structure (i.e. have similar number of edges and neighbouring nodes). However, the implementation of this algorithm has an exponential complexity [13]. While this can be alleviated by terminating the seed algorithm as soon as only a few nodes have been found as opposed to the entire subgraph, it highlights the fact that identifying and eliminating all the possible channels through which information can be leaked is computationally expensive.

The Backstrom et al. [1] seed identification algorithm can also be used at the propagation phase of the de-anonymization algorithm. Although, Backstrom et al. approach is focused on re-identification attacks, we can use the seed identification algorithm and use only the mappings that the algorithm can determine with the most certainty. The propagation algorithm takes the target network as input. The auxiliary network and the initial seed or identification mapping between the two social networks, is found in the previous phase of the propagation algorithm. With each of the iterations, the propagation algorithm considers all the unmapped or unidentified nodes, and selects an arbitrary node in the target network that has not been mapped for re-identification. Then, using the knowledge gained from the mapped nodes, a score is computed for each of the other unmapped nodes in the auxiliary network. This score is essentially the likelihood that the target node corresponds to the unmapped node in the auxiliary network. If a probability of a match is above a certain threshold then this is considered to be the correct mapping and it is added to the mapping between the auxiliary and target network. The score or strength of the match between two nodes is computed by comparing properties of the nodes. One score is computed using the incoming edges into the node and another using the outgoing edges. These are added together to provide a strength rating. The algorithm terminates when it has identified as many nodes as it could possibly have. A theoretical analysis by Narayann and Shmatikov [13] indicate that the running time of this algorithm is proportional to the product of the total number of edges in the two graphs, and the sum of the highest node degree from each graph. This appears to be a feasible running time for small graphs but is problematic for tightly connected social networks.

B. Identifying Group Memberships

Wondracek et al. [19] demonstrate that knowledge of public groups memberships of users can, at least, significantly increase the accuracy of re-identifying an anonymous individual in the network graph. The problem of finding out which group a user belongs to can be done by browser history stealing techniques. A malicious website, if visited by a user, can detect which sites users have visited. History stealing relies on the fact that Internet browsers style links differently depending on whether they have been visited or not. The malicious site uses this to get the browser to signal whether a user has visited a site. In particular, by carefully constructing the Uniform Resource Locator (URL) that is tested to determine whether or not a user has visited a web page, conclusions can be drawn as to whether or not the user is a member of particular group.

In order to demonstrate the effectiveness of this attack, Wondracek et al. [20] analyse the public groups of

the business social network, Xing. Assuming they were able to discover all of the groups that a user is a member of with no uncertainty, then the success of the attack would be notable. However, this assumption is unrealistic because, as mentioned in the previous section, this requires accurately identifying ann the incoming and outgoing egdes from all the nodes in the graph. Instead, only a partial knowledge of group membership is obtained yielding a less accurate prediction of who the particular user is.

C. Combining the Two Attacks

The node re-identification algorithm and knowledge of group memberships are effective attack strategies for de-anonymizing social networks but are impeded by scability. We note however, that by combining both strategies the chances of achieving greater success in de-anonymizing the graph can increase considerably. In order to combine the two attacks, the idea is to use the group membership attack in the initial seed identification stage of the re-identification algorithm. Ideally, the group membership information would be used throughout the re-identification algorithm, but the complexity and scalability of implementing the attack makes it challenging.

A more practical method of extracting group membership information is to lure users in to consenting to visit a site that performs the history steal. However, if this were implemented illicitly each user would need to be tricked into visiting a site that does a history stealing technique, possibly via a phishing attack. This would be very difficult to achieve across a large social network, because of the vigilance of the network operators preventing automated spam. So, it is more practical to get only a few users to visit the history stealing site, which would allow for a small number of users to be identified with high accuracy. In order to generate more realistic messages to attract users to the site that implements a history stealing technique, methods to generate realistic looking messages. This would aid in defeating automated spam detection methods designed by network operators. Additionally, we could develop a third party application (e.g. Farmville on Facebook [5]) that would be a platform to conduct the history stealing technique. By combining this with the original seed finding algorithm, some nodes can be re-identified to move on to the propagation stage.

D. Independent Set Correlation

In graph theory an independent set of vertices in a graph, is a set that is computed such that no two vertices are adjacent. A maximal independent set is an independent set that is such that adding any other vertex to the set causes the set to contain an edge, so in essence to remain independent, the set cannot accept any extra edges. The decisin version of this problem of computing a maximum independent is an NP-Hard optimization problem [3]. Therefore, there is no efficient algorithm for computing the maximum independent set of a graph.

This problem can be easily correlated to the one we are faced with in anonymizing social network graphs. In this case we basically wish to compute an anoymized graph that can be viewed by an outsider as an independent set, where all the vertices in the anonymized graph have no sensitive connections (or edges) between them. As with the maximum independent set problem we want to compute an anonymized social network graph that is complete in the sense that all sensitive edges are removed from the graph. Therefore, the addition of a new sensitive edge should reduce the level of anonymization in the graph thereby alerting the security administrator that a de-anonymization attack is being provoked. Obtaining a completely anonymized social network graph in this way is equivalent to attempting to compute a maximal independent set, and there is no efficient algorithm to do this. We can therefore only use approximation algorithms to provide a best-effort solution to the problem.

V. Conclusions

This paper outlines two categories of techniques for anonymizing published data from social networks. A re-identification attack was presented to illustrate the difficulties of modelling the background knowledge needed by an attacker to create robust and secure anonymization techniques. One of the potential drawbacks in implementing the attack will be instantiating the history stealing attack to determine group membership and linking it to the user account. When conducting this as an experiment, users will have to consent to their accounts being used, as well as having the history attack being performed on them. In this sense, performing this experiment would be easier to control and record than if it were immediately and illegally conducted.

Once the seed finding stage is complete, the success of the propagation stage has to be measured. At first glance it seems that the fraction of the number of de-anonomyised nodes of the total number of nodes is an appropriate measure. However this metric can be easily skewed by the presence of singletons which are basically nodes with no edges since the re-identification algorithm fails on those nodes. In the many online networks the majority of nodes show little or no observable activity. Instead, a better measure is to assign a weight measuring the importance of a node in the network. This importance is quantified using the concept of centrality in which the

sum of the weights of all the re-identified nodes counts as the success of the algorithm.

As future work, we plan to implement these attack techniques on some data drawn from an open source social network like Elgg [4] in order to get publicly available social network data. We will also consider applying an anonymization technique on one of the data sets and attempt the re-identification on the overlap between our social network and the other networks. One particular difficulty that may come up is that of getting large amounts of data of our test social network. Some ways of overcoming it might be to try to find some incentives for encouraging users to test our platform.

Another aspect we plan to look at further is the running time of the seed identification part of the algorithm. This may, in the worst cases, have exponential running times. Terminating when sufficiently many nodes have been identified can, in some cases, ease the time constraint. However, there is the issue of scalability that needs to be dealt with in very large social networks. In order to mitigate this, we plan to use a group membership attack to reduce the scope of the problem and increase the chances of finding potential matches between the overlapping social networks.

Finally, we are currently looking at threat modelling techniques in social network platforms as a method of identifying anonymization issues in social networks and discovering ways of closing the vulnerabilities that provoke de-anonymizations in anonymized social network graphs.

REFERENCES

[1] L. Backstrom, C. Dwork, and J. Kleinberg, "Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography," *Commun. ACM*, vol. 54, no. 12, pp. 133–141, Dec. 2011. [Online]. Available: http://doi.acm.org/10.1145/2043174.2043199

[2] L. Barkhuus and J. Tashiro, "Student socialization in the age of facebook," in *Proceedings of the 28th international conference on Human factors in computing systems*, ser. CHI '10. New York, NY, USA: ACM, 2010, pp. 133–142. [Online]. Available: http://doi.acm.org/10.1145/1753326.1753347

[3] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Third Edition*, 3rd ed. The MIT Press, 2009.

[4] Elgg. (2012) Elgg. web page. [Online]. Available: http://www.elgg.org/

[5] FaceBook. (2012) facebook. www. Facebook. [Online]. Available: http://www.facebook.com/

[6] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: A survey of recent developments," *ACM Comput. Surv.*, vol. 42, no. 4, pp. 14:1–14:53, Jun. 2010. [Online]. Available: http://doi.acm.org/10.1145/1749603.1749605

[7] M. Hay, G. Miklau, D. Jensen, D. Towsley, and C. Li, "Resisting structural re-identification in anonymized social networks," *The VLDB Journal*, vol. 19, no. 6, pp. 797–823, Dec. 2010. [Online]. Available: http://dx.doi.org/10.1007/s00778-010-0210-x

[8] D. Kifer and J. Gehrke, "l-diversity: Privacy beyond k-anonymity," in *In ICDE*, 2006, p. 24.

[9] C. Lampe, N. B. Ellison, and C. Steinfield, "Changes in use and perception of facebook," in *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, ser. CSCW '08. New York, NY, USA: ACM, 2008, pp. 721–730. [Online]. Available: http://doi.acm.org/10.1145/1460563.1460675

[10] LinkedIn. (2012) Linkedin. LinkedIn. [Online]. Available: http://za.linkedin.com/

[11] Mendeley. (2012) Mendeley. Mendeley. [Online]. Available: http://www.mendeley.com/

[12] E. Mills. (2012, July) Searchable facebook user data posted to pirate bay. CNET news. [Online]. Available: http://news.cnet.com/8301-27080_3-20012115-245.html

[13] A. Narayanan and V. Shmatikov, "De-anonymizing social networks," in *Security and Privacy, 2009 30th IEEE Symposium on*, may 2009, pp. 173 –187.

[14] S. Ninghui Li; Tiancheng Li; Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity," in *IEEE 23rd International Conference on Data Engineering, Istanbul, Turkey: IEEE, 2007*, 2007, pp. pp. 106–115.

[15] F. (Statistics). (2012, March) Statistics - facebook. Facebook. [Online]. Available: http://newsroom.fb.com/content/default.aspx?NewsAreaId=22

[16] J. Stenros, J. Paavilainen, and J. Kinnunen, "Giving good 'face': playful performances of self in facebook," in *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, ser. MindTrek '11. New York, NY, USA: ACM, 2011, pp. 153–160. [Online]. Available: http://doi.acm.org/10.1145/2181037.2181062

[17] Twitter. (2012) Twiiter. Twitter. [Online]. Available: https://twitter.com/

[18] Y. Wang, G. Norcie, S. Komanduri, A. Acquisti, P. G. Leon, and L. F. Cranor, ""i regretted the minute i pressed share": a qualitative study of regrets on facebook," in *Proceedings of the Seventh Symposium on Usable Privacy and Security*, ser. SOUPS '11. New York, NY, USA: ACM, 2011, pp. 10:1–10:16. [Online]. Available: http://doi.acm.org/10.1145/2078827.2078841

[19] G. Wondracek, T. Holz, n. Kirda, Engi, and C. Kruegel, "A practical attack to de-anonymize social network users," in *Security and Privacy (SP), 2010 IEEE Symposium on*, may 2010, pp. 223 –238.

[20] R. C.-W. Wong, J. Li, A. W.-C. Fu, and K. Wang, "k-anonymity: an enhanced k-anonymity model for privacy preserving data publishing," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '06. New York, NY, USA: ACM, 2006, pp. 754–759. [Online]. Available: http://doi.acm.org/10.1145/1150402.1150499

[21] E. Zheleva and L. Getoor, "Preserving the privacy of sensitive relationships in graph data," in *KDD Workshop on Privacy, Security, and Trust in KDD (PinKDD) 2007, LNCS, vol. 4890*, 2007, pp. 153 –171.

[22] B. Zhou, J. Pei, and W. Luk, "A brief survey on anonymization techniques for privacy preserving publishing of social network data," *SIGKDD Explor. Newsl.*, vol. 10, no. 2, pp. 12–22, Dec. 2008. [Online]. Available: http://doi.acm.org/10.1145/1540276.1540279