Hiding Information in Social Networks from De-anonymization Attacks by Using Identity Separation

Gábor Gulyás, Sándor Imre

Hiding Information in Social Networks from De-anonymization Attacks by Using Identity Separation

Gábor György Gulyás and Sándor Imre

Department of Networked Systems and Services,
Budapest University of Technology and Economics,
Magyar tudósok krt. 2., H-1117 Budapest, Hungary
{gulyasg,imre}@hit.bme.hu

Abstract. Social networks allow their users to mark their profile attributes, relationships as private in order to guarantee privacy, although private information get occasionally published within sanitized datasets offered to third parties, such as business partners. Today, powerful de-anonymization attacks exist that enable the finding of corresponding nodes within these datasets and public network data (e.g., crawls of other networks) solely by considering structural information. In this paper, we propose an identity management technique, namely identity separation, as a tool for hiding information from attacks aiming to achieve large-scale re-identification. By simulation experiments we compare the protective strength of identity management to the state-of-the-art attack. We show that while a large fraction of participating users are required to repel the attack, with the proper settings it is possible to effectively hide information, even for a handful of users. In addition, we propose a user-controllable method based on decoy nodes, which turn out to be successful for information hiding as at most 3.33% of hidden nodes are revealed in our experiments.

Keywords: social networks, privacy, de-anonymization, identity separation

1 Introduction

The basic concept of online social networks is to provide an interface for managing social relationships. However, social networks are not the only services that have an underlying graph structure, and recently several network alignment attacks have been published in which attackers aimed to breach the privacy of nodes within anonimized networks (e.g., obtained for business or research purposes) by using data from other (social) networks [1–4, 11]. Basically, such attacks can have two goals, i.e., to achieve node or edge privacy breach (or both). In case of the first, the attacker learns the identity of a node, or some otherwise hidden profile information, and in the second case the attacker realizes the existence of a hidden relationship.

The first attack of its kind was introduced by Narayanan and Shmatikov in 2009 [1], who proposed a structural re-identification algorithm being able to de-anonymize users at large-scale, by using data from another social network. In their main experiment they de-anonymized 30.8% of nodes being mutually present in a Twitter and a Flickr

crawl. Recently it has been shown that location information can also be re-identified with similar methods [4]. As there are many services based on the graph structure (or implicitly having one), it is likely that more similar attacks will be discovered.

Attacks capable of achieving large-scale re-identification consist of two sequential phases, the global and local re-identification phase [9]. In the first phase the algorithm seeks for globally outstanding nodes (called the seeds), e.g., by their degree, and then the second phase extends the seed set in an iterative way, locally comparing nodes being connected to the seed set.

For instance, an attacker may obtain datasets as depicted on figure 1, wishing to know an otherwise inaccessible private attribute: who prefers tea or coffee (dashed or thick bordered nodes). She initializes the seed set by re-identifying (or mapping) $v_{Alice} \leftrightarrow v_7$ and $v_{Bob} \leftrightarrow v_3$ as they have globally the highest degree in both networks (global re-identification phase). Next, she looks for nodes with locally unique degree values connecting to both seeds, and picks $deg(v_{Harry}) = 3$. By looking for nodes within the common neighbors of v_3, v_7 with the same degree, she maps $v_{Harry} \leftrightarrow v_4$. Then, the process continues with additional iterations.

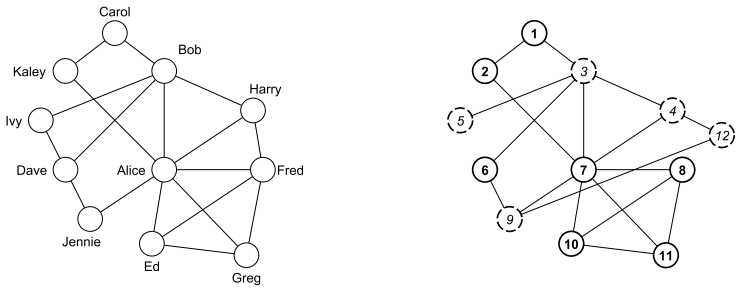


Fig. 1: Example providing insights on de-anonymization and identity separation (left: auxiliary network G_{src}; right: sanitized network G_{tar}).

In this paper we propose a privacy-enhancing method related to the identity partitioning technique [8], called identity separation, to tackle these attacks. Identity separation allows a user to have multiple unlinkable profiles in the same network, which results in multiple unlinkable nodes in the sanitized graph (e.g., as the service provider is also unaware of the link between the identities).

We present effect of identity separation by the example of Fred on Fig. 1, who created two unlinkable profiles, v_8 for pretending being a coffee fan towards his closer friends (Alice, Ed, Greg), but also created v_{12} for maintaining relationships with tea lovers (Harry, Jennie). By applying the attack algorithm, it can be seen that the hidden drink preference of Fred will not be discovered by third parties.

Our main contributions are as follows. By simulation measurements we characterize how resistant the attack in [1] is against different features of identity separation, e.g., splitting nodes or deleting edges. In experiments we show that by using these features the quantity of information revealed by the attacker can be reduced as low as 3.21% and even lower, while identity separation cannot efficiently repel the attack on the network level. We additionally propose a method using decoys that can effectively protect

privacy in a user-controllable way, for which we measure the quantity of revealed information well under 4%, even when only a few users adopt this technique.

2 Related Work

The first attack proposed by Narayanan and Shmatikov in [1] (to which we later refer as Nar09) used seeding of 4-cliques, and its local re-identification phase works similarly as described in the example of Section 1, being based on a propagation step which is iterated on the neighbors of the seed set until new nodes can be identified (already identified nodes are revisited). In each iteration, candidates for the currently inspected source node are selected from target graph nodes, sharing at least a common mapped neighbor with it. At this point the algorithm calculates a score based on cosine similarity for each candidate. If there is an outstanding candidate, a reverse match checking is executed to verify the proposed mapping from a reversed point of view. If the result of reverse checking equals the source node, a new mapping is registered.

However, since then several attacks appeared, here we include the most relevant works. Narayanan et al. in 2011 presented a specialized version of their attack [2], which was capable of achieving a higher recall rate, but was specialized for the task of working on two snapshots of the same network. More recently, Wei et al. also proposed another algorithm in [3] challenging [1]; however, we argue that their work goes beyond [1], for at least two reasons. First, in their paper there is no evaluation of their algorithm against the perturbation strategy proposed in [1], although it is definitely more realistic than what is used in [3]. As the perturbation strategy of [1] deletes edges (only), this is remarkable deficiency. Our second remark is also related to their experiments, which were performed on quite small graphs having fewer than a thousand nodes; there are no experimental results that their algorithm perform also better on networks having tens of thousands of nodes (or larger). Finally, there are some other works developing the original idea further in specific directions, such as in the case for de-anonymizing location traces by Srivatsa and Hicks [4]; however, as to the best of our knowledge no work provides better results than [1] in general, we chose this attack as the state-of-the-art, and work with it in our experiments.

For preventing de-anonymization, we consider a user centered privacy protection mechanism (instead of graph sanitization applied by the service provider), one that can be applied to existing services – otherwise one might consider using revised service models, such as distributed social networks [6]. In our previous work we analytically showed that identity separation is an effective tool against 4-clique based global re-identification [5] (as described later, we use structural identity separation models from this work).

Recently, Beato et al. proposed the friend-in-the-middle model [7], where a proxy-like nodes serve as mediators to hide connections, and presented the viability of their model (successfully) on the Slashdot network [10]. In contrast to their work, we focus also on information hiding working even for a few nodes only, and in addition, identity separation is a rather powerful method allowing a fine-grained management of information [8], e.g., it allows hiding profile information beside relationships. Lastly, we note

that as network structure has a notable bias on results, we carry out experiments on multiple datasets.

3 Datasets, Modeling and Simulation Settings

We partially base our notation on the one used in [1]. Given a graph G_{tar} to be de-anonymized by using an auxiliary data source G_{src}, let $\tilde{V}_{src} \subseteq V_{src}, \tilde{V}_{tar} \subseteq V_{tar}$ denote the set of nodes mutually existing in both. Due to the presence of nodes using identity separation, ground truth information is represented by two mappings, $\mu_G : \tilde{V}_{src} \to \tilde{V}_{tar}$ denote mapping between nodes that are intact, and $\lambda_G : \tilde{V}_{src} \rightrightarrows \tilde{V}_{tar}$ denote mappings between nodes in G_{src} and the sets of their separated identities in G_{tar}. Running a deterministic re-identification attack on (G_{src} , G_{tar}) results in a re-identification mapping denoted as $\mu : V_{src} \to V_{tar}$.

3.1 Social Network Data and Modeling Identity Separation

During our experiments we used multiple datasets with different characteristics in order to avoid related biases. In addition, we used large networks, as brute-force attacks can be mounted against smaller ones. We obtained two datasets from the SNAP collection [10], namely the Slashdot network crawled in 2009 (82,168 nodes, 504,230 edges) and the Epinions network crawled in 2002 (75,879 nodes, 405,740 edges). The third dataset is a subgraph exported from the LiveJournal network crawled in 2010 (at our dept.; consisting of 66,752 nodes, 619,512 edges).

For modeling identity separation, it would be desirable to analyze real-world data on user behavior, but to the best of our knowledge, such datasets are unavailable and there are no trivial ways of crawling one (yet). Fortunately, data on a functionality similar to identity separation is available: structural information of social circles extracted from Google+, Twitter and Facebook [10]. We found in this data that the number of circles has a power-law distribution, for instance in the Twitter dataset we measured $\alpha = 2.31$ (933 ego networks, $x_{min} = 2$, $x_{max} = 18$). Many users did not duplicate their connections (44.6%), and only a fragment of them had more than twice as many connections in their circles compared to the number of their unique acquaintances (6.07%). While it is not possible to draw strong conclusions from these observations, we believe they indicate the real nature of identity separation (the usability of this dataset is limited by the absence of hidden connections).

Thus, due to the lack of data, we used the probability based models we introduced in [5], which describe identity separation from a structural point of view, and allow deriving test data from real-world datasets. These models capture identity separation as splitting a node, and assigning previously existing edges to the new nodes. The number of new identities is modeled with a random variable Y (with unspecified distribution), which we either set to a fixed value, or model it with a random variable having a power-law-like distribution. For edge sorting, there are four models in [5] regarding whether it is allowed to delete edges (i.e., an edge becomes private), or to duplicate edges, from which we used three in our experiments. While the basic model is simple and easy to work with (no edge deletion or duplication allowed), we used the realistic model

to capture real-life behavior, too (both operations are allowed). We additionally used the best model describing a privacy oriented user behavior (no edge duplication, but deletion allowed), and omitted the worst model (edge duplication only).

3.2 Data Preparation

During the test data creation process first we derived a pair of source and target graphs (G_{src}, G_{tar}) having desired overlap of nodes and edges, and then modeled identity separation on a subset of nodes in the target graph. We used the perturbation strategy proposed by Narayanan and Shmatikov [1]. Their algorithm considers the initial graph as the ground truth of real connections, from which graphs G_{src}, G_{tar} are extracted with the desired fraction of overlapping nodes (α_v), and then edges are deleted independently to achieve edge overlap α_e.

We found $\alpha_v = 0.5$, $\alpha_e = 0.75$ to be a good trade-off at which a significant level of uncertainty is present in the data (capturing the essence of a life-like scenario), but the Nar09 attack is still capable of identifying a large ratio of the co-existing nodes[1]. Identity separation is then modeled on the target graph by uniformly sampling a given percent of nodes with at least $deg(v) = 2$ (this ratio is maintained for the ground truth nodes), and then nodes are split and their edges are sorted according to the settings of the currently used model.

3.3 Calibrating Attack Parameters and Measuring Success Rate

By comparing the directed and undirected versions of Nar09, we found little difference in results, therefore, due to this reason and for sake of simplicity, in our experiments we used undirected networks. Next, we run several measurements to find the optimal parameters of the attack. We found choosing randomly a 1,000 from the top 25% (by node degree) of mutually existing nodes to be a redundant choice modeling a strong attacker (as 750 seeds were enough for reaching the high-end of large scale propagation).

Seed location sensitivity of the algorithm is known for small networks [9]. In contrast, we found that seed location matters less for large networks, likely because the greater redundancy in topology against perturbation, and larger ground truth sizes. Therefore, in each experiment we created two random perturbations, and run simulations twice on both with a different seed set. We observed only minor deviations in results, usually less than a percent.

The Nar09 algorithm has another important parameter (Θ) for controlling the ratio of true and false positives. The attack produced fairly low error rates even for small values of Θ, hence we choose to work with $\Theta = 0.01$. The error rate stayed well under 3% in most of experiments, with only a few exceptions when it went above slightly this value.

We use two measures for evaluating simulation results. The *recall rate* reflects the extent of re-identification (this itself can be used due to constantly negligible error

[1] Without adding perturbation Nar09 could correctly identify 52.55% of coexisting nodes in the Epinions graph, 68.34% in the Slashdot graph, and 88.55% in the LiveJournal graph; identification rates were consequently proportional to the ratio of one-degree nodes.

rates), describing success from an attacker point of view. It is calculated by dividing the number of correct identifications with the number of mutually existing nodes (seeds are excluded from the results).

The *disclosure rate* quantifies information the attacker learned from users who applied identity separation, describing an overall protection efficiency from a user point of view. As current identity separation models are bond to structural information, we use a measure reflecting the average percent of edges that the attacker successfully revealed (this can be extended for other types of information in future experiments, e.g., sensitive profile attributes).

4 Characterizing Weaknesses of the Nar09 Algorithm

In the first part of our experiments, in order to discover the strongest privacy-enhancing identity separation mechanisms, we investigated the efficiency of features in different models against the Nar09 algorithm.

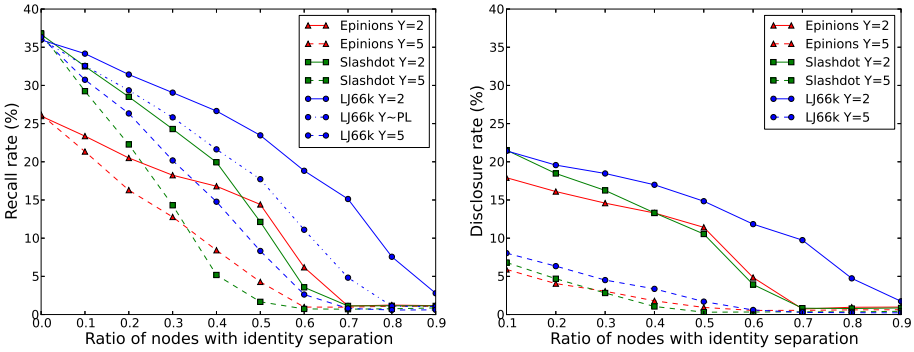
4.1 Measuring Sensitivity to the Number of Identities

Foremost, we tested the Nar09 algorithm against the *basic model with uniform edge sorting probability*. Simulations of the attack were executed for all networks having a ratio of users applying identity separation of $R \in [0.0, 0.9]$ (with stepping 0.1). For the selected users a fixed number of new identities were created ($Y \in [2,5]$). We summarized results on Fig. 2; however, we omitted results for cases of $Y = 3, Y = 4$, as these can be easily inferred from the rest.

Opposing our initial expectations, the basic model with $Y = 2$ and uniform edge sorting probability is not effective in stopping the attack. For the Epinions and Slashdot networks the recall rate mildly decreased until the ratio of privacy-protecting users reached circa $R = 0.5$. For the LiveJournal graph the recall rate shows relevant fault tolerance of the attack (likely because of network structure, as this is also the most dense test network), e.g., Nar09 still correctly identified 15.12% of users even for $R = 0.7$. When participating users had five new identities, results were more promising, as recall rates dropped below 10% at $R = 0.5$ for all networks.

We also tested what if edges are sorted according to power-law distribution having $Y = 5$. These experiments resulted in a slightly higher true positive rate, which is not very surprising: if edges are not uniformly distributed it is more likely for an identity to appear that has more of the original edges than the others (with higher chances to be re-identified). In another comparative experiment we modeled a variable number of new identities with power-law-like distribution with $Y \in [2,5]$ and uniform edge sorting probability. Results were properly centered between cases $Y = 2$ and $Y = 5$ as the LiveJournal example shows on Fig. 2a.

Even though by looking at the recall rates the basic model seems ineffective in impeding the attack, the disclosure rates imply better results. As shown on Fig. 2b, disclosure rates are significantly lower compared to recall rates[2]. From this point of view using the basic model with $Y = 5$ and uniform edge sorting probability provides strong protection for even a small ratio of applying users: the disclosure rate is at most



(a) Recall rates (b) Disclosure rates

Fig. 2: Experimental results using the basic identity separation model.

8.03% when $R = 0.1$. By comparing the results of the two measures, we conclude that by using the basic model it is not feasible to repel the attack, however, by using a higher number of identities the access of the attacker to information can be effectively limited.

While conducting the analysis, we found that the recall rate was notably higher for users of identity separation ($\forall v \in dom(\lambda_G)$) than for others ($\forall v' \in \tilde{V}_{src}$). For low values of R this difference in the recall was almost constant and disappeared for high values[3]. This turned out to be a bias caused by the seeding strategy: after changing to mixed seeding with an equal ratio of seeds selected from $dom(\mu_G)$ and $dom(\lambda_G)$, while the overall recall rate remained equivalent the difference disappeared for the LiveJournal and Slashdot networks, and significantly decreased for the Epinions. Most importantly (from the user perspective), the disclosure rates stayed equivalently low regardless of the used seeding strategy.

This finding has an interesting impact for the attacker on choosing the proper seeding strategy. Using a simple seeding mechanism seems to be a natural choice, but adding fault tolerance against identity separation is not trivial: the analysis provided by in [5] shows that the seeding method discussed in [1] is not very resistant to identity separation. Thus, using a simpler choice of seed identification, the attacker will also have a higher rate of correct identification for nodes protecting their privacy.

4.2 Measuring Sensitivity to Deletion of Edges

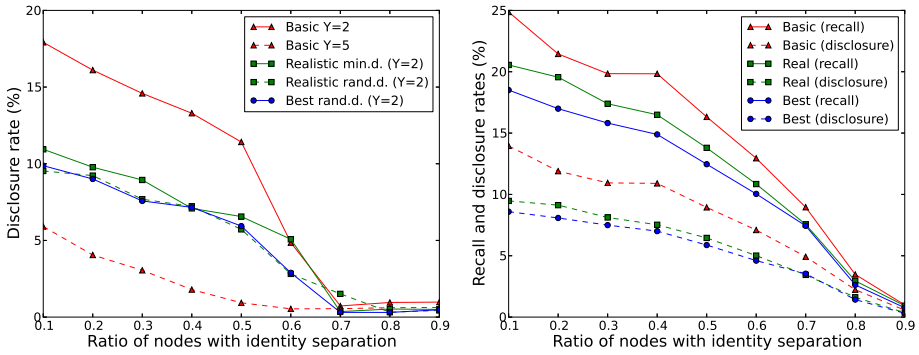
We used the realistic and best models to test the Nar09 against additional edge perturbation [5], as edge deletion is allowed during the edge sorting phase within these models. Since details are not explicitly defined in [5], we used three different settings in our

[2] As the disclosure rate is measured for $\forall v \in dom(\lambda_G)$, results start from $R = 0.1$.

[3] We omit plotting this on a figure due to space limitations, however, the difference was as follows: $avg(\Delta_{Slashdot}) = 2.34\%\ \forall R \in [0.1, 0.4]$; $avg(\Delta_{Epinions}) = 6.13\%\ \forall R \in [0.1, 0.5]$; $avg(\Delta_{LiveJournal}) = 4.05\%\ \forall R \in [0.1, 0.7]$

experiments. For all of them edge sorting probabilities are calculated according to multivariate normal distribution as $P(X_1 = x_1, \ldots, X_y = x_y) \sim \mathcal{N}_y(\eta, \Sigma)$, where y denotes the current number of identities. We set each value of η to $(y)^{-1}$ and configure Σ in a way to have higher probabilities for events when the sum of the new edges are relatively close to original node degree (in the best model when the sum was higher than the original degree, the distribution was simply recalculated).

The first setting is the *realistic model with minimal deletion*, in which each edge is assigned to each identity, and if there is still ample space left, random edges are assigned to those identities. In this setting edges are not deleted if it is not necessary. In the setting of the *realistic model with random deletion* new identities take a portion of edges proportional to $(x_1, \ldots, x_y)$. This setting is expected to delete unassigned edges proportionally to $\prod(1 - x_i)$. We also included a setting with the best model for comparison, namely the *best model with random deletion*[4].



(a) Epinions network (b) LiveJournal network

Fig. 3: (a) Disclosure rates for different models in the Epinions network; (b) multiple models present in parallel in the LiveJournal network.

We ran simulations for all models in all the test networks with $Y = 2$, and found that recall rates strongly correlate with results of the basic model (although being slightly better); thus, these models are also incapable of repelling the attack. Fortunately, disclosure rates show significant progress from the basic model; as an example, results for the Epinions network are depicted on Fig. 3a. We conclude that while these models are also incapable of stopping large-scale propagation, they yet perform better in privacy protection.

4.3 Simulating Multiple Model Settings in Parallel

In previous subsections we described experiments in which settings of different models were used homogeneously. Naturally, the question arises whether the observed differ-

[4] We note, however, that none of these settings capture aggressive edge deletion, but it might be interesting to investigate such settings in the future.

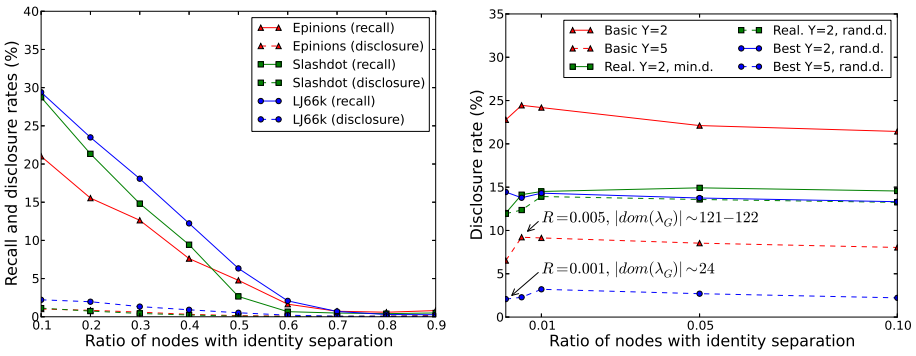
ences remain if multiple settings are allowed in the same network in a mixed way? Hence in another experiment we allowed three settings in parallel: basic model with uniform edge sorting probability (34% of R), realistic model with random deletion (33% of R), best model with random deletion (33% of R). We found that for the users of each setting was proportional to results measured in previous experiments, for instance, users of the best model achieved the lowest recall and disclosure rates. Simulation results in the LiveJournal graph are plotted on Fig. 3b for demonstration (results were measured for homogeneous groups consisting of nodes having the same setting).

5 Searching for Strongest User Protection Mechanisms

5.1 Measuring the Best Trivial Strategies

Previously we characterized weaknesses of the Nar09 attack, and it turned out that while none of the previously analyzed defense strategies can effectively stop it, some forms of identity separation can reduce the amount of accessible information. It also turned out that increasing the number of new identities has a powerful impact on the disclosure rate, while edge perturbation has a less, but yet remarkable effect. Thus, the best model with a high number of identities seems to be the most effective setting.

We run the best model with $Y = 5$ (using the same distribution as described in Section 4) on all test networks. Results revealed even this method cannot prevent large-scale re-identification when a relatively low ratio of users apply the technique. Instead, for all networks the re-identification rate converged to a hypothetical linear line monotonically decreasing as R increase (see Fig. 4a). Fortunately, the setting had more convincing results for disclosure rates: even for $R = 0.1$ the disclosure rate topped at 2.22%. Disclosure values continued to fall as the ratio of defending users increased.



(a) Best model with $Y = 5$.

(b) Low participation rates in LiveJournal.

Fig. 4: Analysis of the most effective privacy-enhancing strategies.

In addition, we also examined disclosure rates for cases when participation were very low such as 1‰ of V_{tar}, meaning only a few tens or hundreds of users using identity separation from $\tilde{V}_{tar}$. As seen on Fig. 4b, our experiments resulted in approximately

constant disclosure rates for all models (variability for $R < 0.01$ is likely to be due to the small sample sizes). Therefore we conclude that even if only a few users use the best model with $Y = 5$, their privacy is protected as the attacker can reveal only a few percent of sensitive information.

5.2 Placing the User in the Position of Decision

Strategy proposed previously lacks user control, i.e., the user cannot influence on what information she wishes to hide from the attacker. Here, we introduce a simple model that puts the user into decisive position by utilizing decoy identities. Nevertheless, we note that this model is a simple example, and ones used in real-life situations can be adapted to other hypothetical attacker strategies and to the type of information for hiding (e.g., one may consider using structural steganography for hiding nodes [11]).

We applied the following strategy on nodes $v_i \in \tilde{V}_{tar}$ that have at least 30 neighbors[5]. First, we create a decoy node v_i^{decoy} representing non-sensitive connections with the goal of capturing the attention of attacker algorithm (this may be a public profile as well). Node v_i^{decoy} is assigned 90% of the acquaintances of v_i. Next, a hidden node v_i^{hidden} is created having the rest 10% of neighbors (i.e., sensitive relationships), and an additional 10% that overlaps with the neighbors of v_i^{decoy} (i.e., modeling overlapping relationships).

This model showed promising results after being applied to our test data sets. While from the attacker point of view the algorithm was successful, as being able to produce high recall rates (until large number of decoys appeared – see details on Fig. 5a), privacy-protecting nodes achieved of revealing little sensitive information as shown on Fig. 5b. Recall rates were typically small for hidden nodes, less or equal 0.6% within the Slashdot and the Epinion networks, and with one exception less or equal 1.66% within the LiveJournal network. Misguidance was also successful when only a few users used it[6].

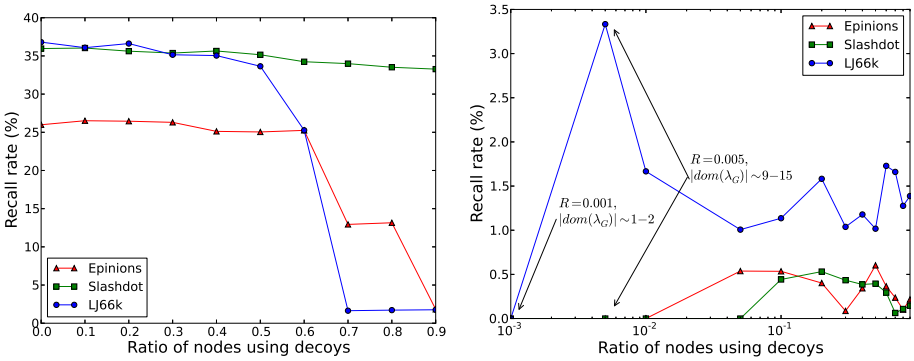
6 Future Work and Conclusions

In this paper, we analyzed different models of identity separation to evaluate their effect in repelling structural de-anonymization attacks and in information hiding. By our experiments we found that if identity separation is used in a non-cooperative way, it is not possible to avoid large-scale re-identification regardless of the used strategy, unless a large fraction of users is involved. This finding sets a direction for future work: is there a way for cooperating users to tackle these attacks more effectively?

We also used another measure in our experiments, reflecting the quantity of information a successful attack reveals. This metric showed more promising results: experiments confirmed that using multiple identities and allowing to hide some connections

[5] Resulting in a significantly smaller set of applicable nodes, e.g., in LiveJournal, even for $R = 0.9$ only $|dom(\lambda_G)| \approx 11.2\%$ of $\tilde{V}_{tar}$.

[6] Appearing variability is due to small sample sizes, and almost negligible. For instance, the recall of 3.33% means one node being identified correctly within 29 cases.



(a) Recall sensitivity to the use of decoys.

(b) Results for nodes with decoys.

Fig. 5: Recall rates for whole networks and nodes using decoy nodes.

favors user privacy. Moreover, in our experiments using five identities with a moderate preference for hiding or duplicating edges proved to be eligible to achieve a high rate of information hiding. Numerically, in the LiveJournal network we measured an information disclosure of 2.08% when only circa 24 users applied the proposed strategy, and yet results were well under 4% in other cases as well.

However, using five identities is not realistic for most users, and in this case it is not possible to control what the attacker may reveal. Therefore, we proposed a method of using decoys, which seemingly did not affect the success of the attack (unless used by more than half of the users); however, as the attacker almost discovered decoy nodes only, a minority of hidden nodes were found: less or equal 0.6% within the Slashdot and the Epinion networks, and with one exception less or equal 1.66% within the LiveJournal network. This method also produced suitable results when applied by a few nodes only (i.e., numerically 1-2).

Therefore, we have provided guidelines (in the form of two models) for effectively realizing information hiding in social networks, that can be applied to existing social networks, even without the consent of the service provider. As our closing word, we designate an interesting direction as future work: what strategies should a user follow, if identity separation can be applied in both networks of G_{src} and G_{tar}?

References

1. Narayanan, A., Shmatikov, V.: De-anonymizing social networks. In: 30th IEEE Symposium on Security and Privacy, pp. 173–187. IEEE Press, New York (2009)
2. Narayanan, A., Shi, E., Rubinstein, B.I.P.: Link prediction by de-anonymization: How we won the kaggle social network challenge. In: The 2011 International Joint Conference on Neural Networks, pp. 1825–1834. IEEE Press, New York (2011)
3. Peng, W., Li, F., Zou, X., Wu, J.: Seed and Grow: An attack against anonymized social networks. In: 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, pp. 587–595. IEEE Press, New York (2012)

4. Srivatsa, M., Hicks, M.: Deanonymizing mobility traces: Using social network as a side-channel. In: 2012 ACM conference on Computer and communications security, pp. 628–637. ACM Press, New York (2012)
5. Gulyás, G.Gy., Imre, S.: Analysis of Identity Separation Against a Passive Clique-Based De-anonymization Attack. Infocommunications Journal, III(4), 11–20 (2011)
6. Cutillo, L.A., Molva, R., Strufe, T.: Safebook: a Privacy Preserving Online Social Network Leveraging on Real-Life Trust. IEEE Communications Magazine, 47(12), 94-–101 (2009)
7. Beato, F., Conti, M., Preneel, B.: Friend in the Middle (FiM): Tackling De-Anonymization in Social Networks. 5th IEEE International Workshop on Security and Social Networking (2013)
8. Clauß, S., Kesdogan, D., Kölsch, T.: Privacy enhancing identity management: protection against re-identification and profiling. In: 2005 workshop on Digital identity management, pp. 83–94. ACM Press, New York (2005)
9. Gulyás, G.Gy., Imre, S: Measuring Local Topological Anonymity in Social Networks. In: 12th International Conference on Data Mining Workshops, pp. 563–570. IEEE Press, New York (2012)
10. Stanford Large Network Dataset Collection, `http://snap.stanford.edu/data/index.html`
11. Backstrom, L., Dwork, C., Kleinberg, J.: Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In: 16th international conference on World Wide Web, pp. 181–190. ACM Press, New York (2007)