Measuring Importance of Seeding for Structural De-anonymization Attacks in Social Networks

Gábor György Gulyás
Laboratory of Cryptography and Systems Security,
Budapest University of Technology and Economics, Hungary
Email: gulyas@crysys.hu

Sándor Imre
Mobile Communications and Quantum Technologies Lab.,
Budapest University of Technology and Economics, Hungary
Email: imre@hit.bme.hu

***Abstract*—Social networks allow their users to make their profiles and relationships private. However, in recent years several powerful de-anonymization attacks have been proposed that are able to map corresponding nodes within two seemingly unrelated datasets solely by considering structural information (e.g., crawls of public social networks and datasets published after sanitization). These algorithms consist of two parts: initial selection of seed nodes and then a propagation phase. In related papers, several seeding procedures are proposed, although detailed comparison is often left unexplored, i.e., how one method differs from the others with respect to the overall outcome of the algorithm. In this paper, beside discussing the existing analysis of seeding methods, we experimentally analyze how different seed selection algorithms perform compared to each other, and we highlight significant differences emerging even in the same or in structurally divergent networks.**

***Index Terms*—Privacy, Re-identification, Simulation, Social Networks.**

I. Introduction

Social network based services have a wide variety of functionality. Some provide interfaces for managing social relationships, while others provide utilities for collaboration. However, a common feature of these services that they have an underlying graph structure which can be used in several useful ways. This feature can also be abused: malicious parties can decide to correlate user identities between networks. Other actors having access to sanitized copies of networks (e.g., business contractors or research groups) can try to reassign original node identities in order to use anonymously published private data without limitations (e.g., data monetization).

The first attack of this kind was the structural de-anonymization attack proposed by Narayanan and Shmatikov in 2009 [1] (Nar09), designed specifically for re-identifying a significant fraction of nodes in large networks. The authors in their main experiment re-identified 30.8% of nodes being mutually present in a Twitter and a Flickr crawl with a relatively low error rate of 12.1%. Since their work, several attacks with the same principles have been published [2]–[7].

These attacks differ in many aspects, however, in general they consist of two sequentially executed phases, namely the global and local re-identification phases [8], or seed identification and propagation phases [1]. The goal of the first phase is to find globally outstanding nodes (the seeds), e.g., by their degree, as an initialization of the second phase. After having a sufficient number of seeds, the second phase starts to extend the seed set in an iterative way, locally comparing nodes being connected to the seed set.

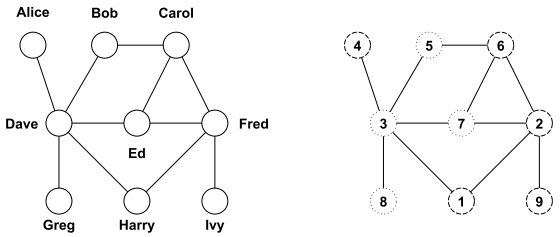


Fig. 1: Datasets for the example of de-anonymization (left: auxiliary public network; right: sanitized network with private attributes).

Let us consider an attacker, Mallory, obtains datasets as depicted on Fig. 1, wishing to learn an otherwise inaccessible private attribute by structural de-annymization: who is a democrat or republican voter in the public network (nodes with dashed or dotted border). Thus, in the global re-identification phase, he creates the seed set by re-identifying (or mapping) $v_{Dave} \leftrightarrow v_3$ and $v_{Fred} \leftrightarrow v_2$ as they have globally the highest (unique) degree values in both networks. Then Mallory continues with local re-identification by inspecting nodes related to the seed set. First he picks v_{Ed} being the common neighbor of the seeds with the highest degree, by comparing and mapping it as $v_{Ed} \leftrightarrow v_7$. Then the algorithm continues iterating through unmapped nodes.

Although multiple adaptations exists of the Nar09 algorithm [2]–[7], and other works use the attack for simulation evaluation of privacy-enhancing features [8]–[10], an important aspect of the attacker model is often neglected: how changing the seeding method influence the performance of the propagation. In our work we aim filling this gap by analyzing multiple methods on different networks, and also including related works discussing this topic [1], [5], [8].

Our main contribution in this paper is that we show that various seeding methods have a different effect on propagation even in the same dataset, but also in structurally divergent

networks. This makes accurate comparison of new attacks and protection mechanisms cumbersome. With running simulations of the original algorithm Nar09, we show that the overall recall rate is influenced by several properties of seeding. One of the most important factors is the measure used for globally selecting the seed set (e.g., betweenness centrality), and the connection between the nodes (e.g., having a cliquish structure). On multiple datasets we show that the number of seeds can also determine the outcome of propagation. The minimum number of nodes required vary per seeding method and network (where size and structure both matter). We give examples how the phase transition attribute of propagation changes for different seeding methods (i.e., when small increments in the number of seeds greatly boost propagation). We furthermore discuss stability of seeding for given seed set sizes, when the given seed set size results wide-scale propagation only occasionally.

II. Related Work

In the original experiment the Nar09 attack used 4-cliques of high degree nodes as seeding. Its local re-identification phase works similarly as described in the example of Section I, 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, the new mapping is registered.

Here we include the most relevant works appeared since [1]. Narayanan et al. in 2011 presented another variant of their attack [2] specialized for the task of working on two snapshots of the same network (with a higher recall rate). Another proposal of Wei et al. [3] challenged Nar09; however, their attack is only evaluated against a light edge perturbation procedure, instead of the more realistic one proposed in [1]. The latter deletes edges from both networks (e.g., node and edge overlaps can be as low as 25%), while in [3] edges are only added to the target network (up to 3%) without deletion; this is a remarkable deficiency. In addition, experiments in [3] are performed on rather small graphs: further experiments need to show if algorithm in [3] also performs better on networks having tens of thousands of nodes or larger (if their attack is still feasible on such datasets).

Recently, it has been shown by Srivatsa and Hicks that location traces can also be re-identified with similar methods [4]. In their work they succeeded in identifying 80% of users by building anonymous networks by observing location traces, and using explicit social networks for de-anonymization. Besides structural re-identification attacks, some works extend the capabilities of structural de-anonymization by involving user content and attributes, too [6], [7], [11].

Seeding is an important aspect of the de-anonymization procedure, as shown by our results. It is needed to be detailed both for comparing new attack schemes and for evaluating protection mechanisms. However, only a few papers discuss the relevance of seeding, and in others, related details are absent. For instance, Narayanan and Shmatikov describe how they used 4-clique seeding consisting of high degree nodes [1], but in another work [3], it is not detailed in how seeds were selected during the evaluation of the propagation phase (i.e., the nodes that the injected subgraph is connected to). Similarly, protection mechanisms as [9], [10] should be evaluated against attackers capable of using multiple seeding methods.

Related to the effect of seeding on propagation, Narayanan and Shmatikov highlight that seeding has a phase transition property regarding the number of seeds [1]: at some point while increasing the number of seeds, there is only a little difference when the output of propagation rises significantly, reaching the maximum (examples provided on Fig. 3). They also note (without details) that transition boundaries depend from networks structure and seeding method. Seeding stability is also mentioned in their paper as the probability of large-scale propagation with respect to the number of seeds.

Yartseva and Grossglauser provide further analysis of seeding [5], and they propose two simpler, but similar algorithms to Nar09, that allow formal analysis. In their work, the existence of phase transition is formally proven w.r.t. to the seed set size for random graphs generated by the Erdős-Rényi model $G(n, p)$. Phase transition is also verified by simulations both for synthetic and real-life social networks. However, their work discusses the essential seed set size for propagation as a function of the network parameters and the propagation settings, but neglects how seeds were obtained, i.e., the seed selection method.

In [8] phase transition feature is verified for small networks, and seed location sensitivity of the algorithm is also asserted. It must be noted however, the seed location is less of an issue for large networks, as it is easier to find enough seeds for a stable output. This is likely to be caused by the greater redundancy in topology against perturbation, and larger ground truth sizes. Although, as shown in our experiments later, seed node degree is still an influential factor of the final output.

While seed size and phase transition are studied aspects of the attacker model in the literature, there are still several questions left open. For instance, how strong is the difference between different seeding methods, e.g., w.r.t. minimum seed size and seeding time? Is there a globally best seeding method? In our work we analyze seeding to answer these and other related questions.

III. Evaluation Method

Given a sanitized graph G_{tar} to be de-anonymized by using an auxiliary data source G_{src} (where node identities are known), let $\tilde{V}_{src} \subseteq V_{src}, \tilde{V}_{tar} \subseteq V_{tar}$ denote the set of nodes mutually existing in both. Ground truth is represented by mapping $\mu_G : \tilde{V}_{src} \rightarrow \tilde{V}_{tar}$ denoting relationship between coexisting nodes. Running a deterministic re-identification

attack on (G_{src} , G_{tar}) initialized by seed set $\mu_0 : V_{src} \to V_{tar}$ results in a re-identification mapping denoted as $\mu : V_{src} \to V_{tar}$.

A. Data Preparation

During our experiments we used multiple datasets with different structural characteristics in order to avoid related biases in the results. In addition, we used large networks consisting of tens of thousands of nodes, where brute-force attacks are not feasible. We obtained two datasets from the SNAP collection [12], 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 (LJ66k) is a subgraph exported from the LiveJournal network crawled in 2010 (at our dept.; consisting of 66,752 nodes, 619,512 edges), and for comparison a smaller dataset (LJ10k) was also included (10,056 nodes, 231,416 edges). All datasets were obtained from real networks in order to maintain our measurements being realistic.

For data generation we used the perturbation strategy proposed by Narayanan and Shmatikov [1], as we found this method to be producing fairly realistic test data. Their algorithm takes the initial graph (μ_G) to derive G_{src}, G_{tar} with the desired fraction of overlapping nodes (α_v), and then edges are deleted independently from the copies 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 (thus life-like), but the Nar09 attack is still capable of identifying a large ratio of the co-existing nodes. Without adding perturbation it 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 (LJ66k). These rates were proportional to one degree nodes, reflecting significant structural differences.

B. Simulation Settings

Our experiments were run on a 2.0GHz Intel Core i7 processor with 8GB RAM, and our framework was implemented in Java. However, we must note that our intent is to show differences between measures and highlight trends, and not to provide razor-sharp results; thus we caution drawing conclusions from subtle differences in results (e.g., minimum seed set sizes of `betwc.1` and `betwc.25` in Slashdot on Fig. 5b).

For each experiment we created two random perturbations, and run the algorithm three times. We found this to be a good trade-off between computation time and having reliable results. In addition, as we found little difference in the results between the directed and undirected versions of Nar09, for the sake of simplicity we used the undirected variant.

Nar09 has another important parameter, denoted as Θ, that controls the ratio of true positives (recall rate) and false positives (error rate). The lower Θ is the less accurate mappings Nar09 is willing to accept, as Θ controls how outstanding the best candidate should be from the others. The attack produced fairly low error rates even for small values of Θ (see Fig. 2), hence we worked with $\Theta = 0.01$. The error rate stayed around 1-2% for large networks, always less 5%; error rate was only proportionally higher in LJ10k to the recall rate.

We also executed experiments for characterizing phase transition prior to our evaluation. For measuring sensitivity of the number of seeds, we executed multiple measurements by selecting random nodes from the 25% of top degree nodes. Our measurements verified the phase transition effect and showed the structure dependency of this property (see Fig. 3).

IV. Seeding Methods

The seeding method reflects the strength of the attacker, who is often limited by the quality of the background knowledge he has. However, a well-informed attacker may have the opportunity to choose between different seeding methods.

A. Used in the Literature

The original paper used high-degree nodes for seeding that formed 4-cliques [1] (in their main experiment they used seed nodes with at least a degree of 80), while another work used nodes from 4-cliques regardless of degree [8] for smaller networks. Several other seeding methods appeared in the literature, as matching top nodes [2], [9], (presumably) sampling random nodes in [3], and seeds selected randomly from top 25% high degree nodes [10].

Srivatsa and Hicks adopted the concept of Nar09 to a special application, namely to matching a social network (G_{src}) with a contact graph of devices (G_{tar}). In their work they used betweenness centrality for seed selection in G_{src} and proposed a probabilistic variant of a distance measure to find corresponding nodes in G_{tar} [4].

B. Seeding Algorithms for Evaluation

In our experiments we generalized clique based methods, where seed nodes were requested to form *k-cliques* ($k \in \{4, 5, 6\}$). We had cases where node degree was not considered (later referred to as e.g., `4cliques`), while in other cases seeds were sampled from the top 20% by degree (e.g., `4cliques.2`). In order to see the magnitude of the effect of the clique structure, we compared these results against *k-neighborhood* seeding (with corresponding parameters), where nodes are collected with breadth-first search starting from a random node (e.g., `4bfs`, `6bfs.2`).

These tests alone could reveal the sensitiveness of the propagation algorithm regarding node degree; however, in order to see how degree itself influence overall results, we included using *k-top* degree nodes (`top`), and sampling from *random high degree nodes* in the top $10\%, 25\%, 50\%$ subsets (e.g., `random.25`), and from all nodes (`random`), for the sake of completeness.

We also analyzed more complex measures than node degree, namely *betweenness* (e.g., `betwc.2`, seeds that had the highest betweenness in the set of the top 20% by degree) and *closeness centrality* (e.g., `closec.2`). These measures can be calculated together as being based on shortest paths: betweenness reflects centrality respecting the number of shortest paths

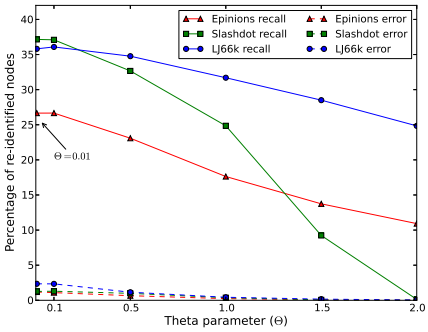


Fig. 2: Varying the Θ parameter on perturbed networks (with `random.25`).

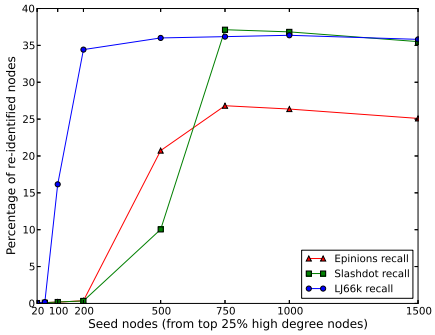


Fig. 3: Phase transition property illustrated for `random.25`.

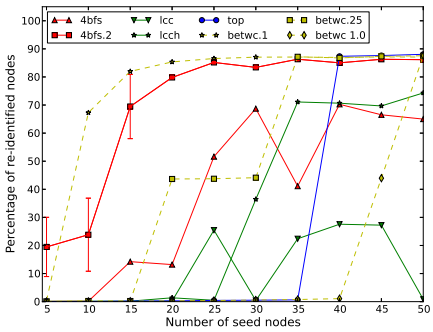


Fig. 4: Differing characteristics of seeding strategies in LJ10k.

the node is on, while closeness gives the average distance from all other nodes in the network. Betweenness was used only for small networks in [4], thus its utility in larger networks was uncertain until this paper. In addition, calculating betweenness and closeness is very costly for large networks, hence we also analyzed if the number of nodes involved in the calculation process can be decreased.

We included two additional exotic seeding measures. *Local Topological Anonymity* (LTA) values are calculated according to the structural uniqueness of nodes in their 2-neighborhoods [8] (the lower the value the more unique the node is), thus our intuition was that nodes with low LTA values are likely to be good seeds (marked as `lta`). We also tested seeding with nodes having the highest *Local Clustering Coefficient* (LCC) values in the network (`lcc`). We had the intuition that probably not the nodes with the most dense neighborhood are providing the better seeds, hence we measured high LCC (`lcch`), where highest LCC nodes were selected after skipping the top 20% of LCC.

V. Evaluation and Results

During our evaluation we calculated measures on G_{src}, and to keep our focus on the comparison, we used the ground truth to map selected seed nodes to their pairs in the G_{tar}. However, these methods can be implemented to work without background knowledge, there are several examples of such implementations in the literature [1], [2], [4].

We were looking to find the minimum number of seeds always granting large-scale propagation in our experiments (i.e., stable seeding) and measured runtime of the seed selection phase (or resource requirements in other words). Although we found only minor differences in recall rates, analyzing further aspects seems to be interesting future work (e.g., run time differences of the propagation phase). We used seed size stepping granularity as 5 in simulations executed on LJ10k and 60 in larger networks (or lower for competitive techniques in order to get more detailed measurements).

It must be noted regarding runtimes that some measures require significant preliminary calculations to seeding: betweenness and closeness centrality (these can be calculated in parallel), LTA, and LCC. We did not include these preparations into seed timings, as although they may run longer, yet these are still computationally feasible (e.g., within a few hours of computation time), and need to be done once. Nevertheless, an attacker may consider this when choosing the seeding method.

A. Large-Scale Propagation

Initial simulations were performed on LJ10k, the smallest network included in our experiments. Results revealed that node degree takes an important position as a secondary measure of seed selection. For all k-clique and k-neighborhood based methods we observed that when using high degree seeds, less nodes are needed for large-scale propagation, and more importantly, Nar09 was able to access the network more widely. For instance, compare `4bfs` and `4bfs.2` on Fig. 4 – there is a clear limit for propagation when using `4bfs` seeding. Thus we used only high-degree variants of the k-clique and k-neighborhood seeding methods in our analysis related to larger networks (in addition, this speeds up seeding).

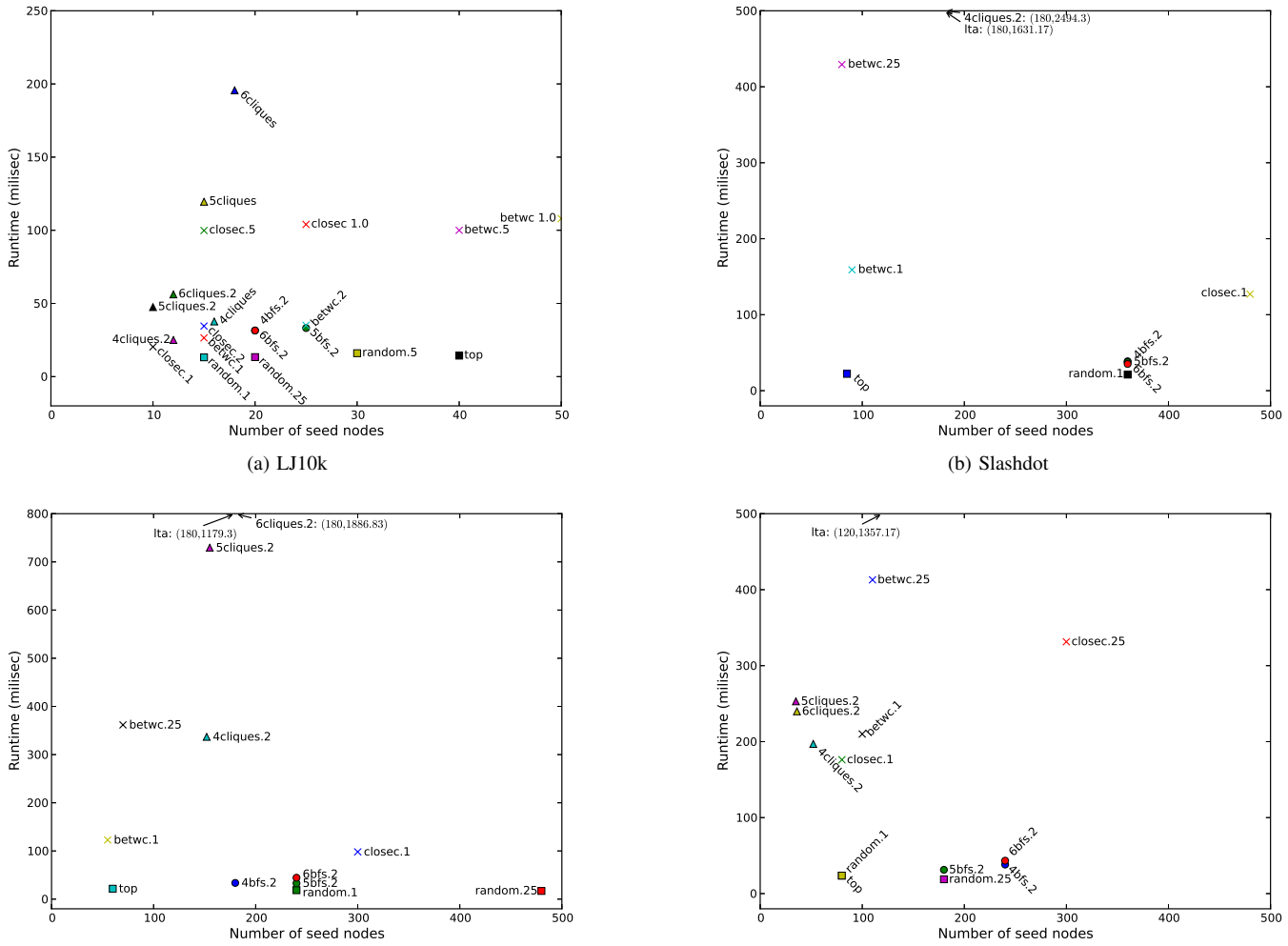
Degree dependent node selection for other measures also lead to differences in results, although it did not limit the maximum level of propagation. The examples shown for betweenness centrality on Fig. 4 illustrate how degree defines the number of seed nodes that are required for successful propagation.

Other factors can influence results, too. While `lcc` could not reach an acceptable level of re-identification in our measurements (resulting recall rates at most around 20%), the `lcch` variant produced better rates, though it was also incapable of reaching recall rate significantly higher than 70%, similarly to `4bfs` (check on Fig. 4).

Additionally, our measurements on Fig. 4 confirm that phase transition property of the propagation phase depends on the seeding measure (as also stated in [1]): phase transition start- and endpoints, steepness differ for various methods. For example, while phase transition both for `4bfs` and `4bfs.2` start early, and have a mild increase, for the `top` method it can be rather characterized as a sharp jump.

B. Seed Stability

Example for `4bfs.2` on the LJ10k network provides insight on seeding stability on Fig. 4. While it allows propagation reaching high-end of recall for $|\mu_0| \in [20, \ldots, 50]$, it



(a) LJ10k

(b) Slashdot

(c) Epinions

(d) LJ66k

Fig. 5: Performance of propagation phase vary as different seeding methods are used. While some methods performed equally well in all cases (e.g., `betwc.1`), some methods produced different results according to the size of the network (e.g., `top`), to structural differences (e.g., `4cliques.2`), or according to the relationship between seed nodes (e.g., `4cliques.2` vs. `4bfs.2`).

can even achieve an average recall of 20% for 5 seed nodes. By including the variance besides (divided by 10^2), a notable variance can be noticed initially, taking values between 1000-1300. As seeding gets stable, it apparently disappears as it takes values between 0.1-23.3 for $\mu_0 > 15$ (other experiments showed similar behavior, but not displayed for keeping the figure clear). This happens for a simple reason: in case of such a small amount of seeds the current instance of seed nodes determines significantly the overall outcome of Nar09, e.g., in these experiments propagation achieved recall rates of 0.26% or 78.1% for different seed sets.

As the error rate is low by design, an attacker can settle with a low number of seeds that leads to large-scale propagation. This even works in larger networks: with only a single 5-clique seeding (`5cliques.2`) we could achieve recall rate as high as 84.33% having the error rate at 5.62%. As future work, it is interesting to check whether currently investigated trends are the same for unstable seeding.

C. Degree as a Heuristic

The summary of our measurements for LJ10k and the other three networks is shown on Fig. 5, including methods that resulted in large-scale propagation, and where runtimes and the number of required seeds were sufficiently low. With accordance of the results in LJ10k, where these measures with higher degree nodes resulted better recall rates, we only calculated betweenness and closeness centrality on high-degree nodes to reduce runtimes (top $10\%, 25\%$). For all three networks results showed that the higher degree nodes we used, the lower the seeding time was.

D. k-cliques and k-neighborhoods

The network structure determines which seeding methods could be used or not. Using cliques were not feasible in the Slashdot network: although it was possible to find enough seeds with `4cliques.2`, this was a less prominent result. In addition, for `5cliques.2` and `6cliques.2` our seeding algorithm timed out (2 mins) before finding enough disjoint cliques. These methods were more competitive in the Epinions network, we measured best results in the LJ66k dataset (the most dense one), as these were capable of stable seeding with the least number of seeds.

For reaching unstable large-scale propagation in our dense test networks, cliques provided also very competitive results. A single clique (of any size) was enough in LJ10k, and we could achieve the highest recall level by simply using two cliques. In LJ66k, a single `4cliques.2` was enough to reach recall of 33.32% with an error rate of 4.34%. Comparing these results to `4bfs.2`, `5bfs.2` and `6bfs.2` shows that structure between seed nodes can make a perceptible difference in the performance of propagation. In addition, the latter techniques were not sensitive to network structure: these had low runtimes in all large test networks, but also had an average score regarding seed sizes.

E. Most Effective Methods

Clearly `top` and `betwc.1` seeding methods led to best results, that were additionally independent of network structure (in larger networks). The discovery of `betwc.1` in this context is important, e.g., as a protective method may aim preventing de-anonymization by targeting top nodes, either by removing or modifying them. Thus `betwc.1` allows the attacker choosing seeds from a larger candidate set. The `random.1` method is slightly less effective, but it could also be used alternatively. Additionally, `closec.1` provided remarkably good results in the densest test network.

F. Exotic Seeding Measures

None of the exotic seeding methods could be put to the front of the ranking. Regarding the minimum number of nodes required for stable seeding, the `lta` measure produced fair results in large networks, but due to the large number of nodes it worked with it had high runtimes. The `lcc` and `lcch` seeding methods had even worse results; both only led to noticeable propagation in LJ66k, and had long runtimes. However, we could not include these results as their highest recall rate was less then the maximum (as in LJ10k).

VI. Conclusion

In this paper we analyzed the effect of the seeding phase on propagation in de-anonymization attacks, and evaluated multiple seed selection methods on the algorithm proposed by Narayanan and Shmatikov [1]. We showed that the chosen method can significantly influence and limit the possible outcome of the propagation. With experiments we showed that both the global role of the seed nodes (measured with betweenness, closeness, degree) and the local structure between them (clique structure vs. k-neighborhood) can solely and jointly determine the success of propagation with the given seeding.

We confirmed and showed examples of phase transition with respect the number of seeds, and also that this attribute has different characteristics for various seeding methods, beside being dependent on network size and structure. However, our work also indicate that the seeding procedure should be chosen regarding network size and structure, as not all methods worked equally well for all datasets. We also highlighted `betwc.1` and `top` that were top performers on the large networks in our experiments, regardless of network structure.

We believe our findings are essential for works aiming to compare novel attack techniques to others and for papers including simulation evaluations of defense methods. For the prior, it is needed to synchronize attacker models, including the seeding method in order to settle down on the same ground for comparing results. In the latter case, seeding methods represents another aspect of the attacker model that can be tuned for alternative (and stronger) attacks. For example, an attacker can react by choosing another seeding procedure in order to decrease the performance of the users of a given privacy-enhancing technique.

Acknowledgment

The authors would like to thank Tamás Holczer and Márk Félegyházi for reviewing drafts of this paper, and Levente Buttyán for discussions on the topic.

References

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

[2] A. Narayanan, E. Shi, and B. I. P. Rubinstein, "Link prediction by de-anonymization: How we won the kaggle social network challenge," in *IJCNN*, 2011, pp. 1825–1834.

[3] W. Peng, F. Li, X. Zou, and J. Wu, "Seed and grow: An attack against anonymized social networks," in *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2012 9th Annual IEEE Communications Society Conference on*, 2012, pp. 587–595.

[4] M. Srivatsa and M. Hicks, "Deanonymizing mobility traces: using social network as a side-channel," in *Proceedings of the 2012 ACM conference on Computer and communications security*, ser. CCS '12. New York, NY, USA: ACM, 2012, pp. 628–637. [Online]. Available: http://doi.acm.org/10.1145/2382196.2382262

[5] L. Yartseva and M. Grossglauser, "On the performance of percolation graph matching," in *Proceedings of the first ACM conference on Online social networks*, ser. COSN '13. New York, NY, USA: ACM, 2013, pp. 119–130. [Online]. Available: http://doi.acm.org/10.1145/2512938.2512952

[6] S. Bartunov, A. Korshunov, S.-T. Park, W. Ryu, and H. Lee, "Joint link-attribute user identity resolution in online social networks," in *Proceedings of the sixth Workshop on Social Network Mining and Analysis*, 2012.

[7] P. Jain, P. Kumaraguru, and A. Joshi, "@i seek 'fb.me': identifying users across multiple online social networks," in *Proceedings of the 22nd international conference on World Wide Web companion*, ser. WWW '13 Companion. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee, 2013, pp. 1259–1268. [Online]. Available: http://dl.acm.org/citation.cfm?id=2487788.2488160

[8] G. G. Gulyás and S. Imre, "Measuring local topological anonymity in social networks," in *Data Mining Workshops (ICDMW), 2012 IEEE 12th International Conference on*, 2012, pp. 563–570.

[9] F. Beato, M. Conti, and B. Preneel, "Friend in the middle (fim): Tackling de-anonymization in social networks," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on*, 2013, pp. 279–284.

[10] G. G. Gulyás and S. Imre, "Hiding information in social networks from de-anonymization attacks by using identity separation," in *Communications and Multimedia Security*, ser. Lecture Notes in Computer Science, B. Decker, J. Dittmann, C. Kraetzer, and C. Vielhauer, Eds. Springer Berlin Heidelberg, 2013, vol. 8099, pp. 173–184.

[11] D. Chen, B. Hu, and S. Xie, "De-anonymizing social networks," 2012.

[12] "Stanford network analysis platform (snap)," http://snap.stanford.edu/, accessed: 2013-10-14.