Wherefore Art Thou R3579X? Anonymized Social Networks, Hidden Patterns, and Structural Steganography by Backstrom, Dwork, & Kleinberg

Daniel Jackoway

May 21, 2014

1 Introduction

Large-scale social networks have emerged as a major part of our world in the last two decades. With it has come research interest into the structure of these graphs.

However, one barrier to research is access to data. Some might suggest that major social networks could release an anonymized social graph that gives the perfect structure of the graph and simply redacts the identification of which account is represented by each node in the graph.

This paper explores a number of attacks on this model and shows that there are many ways to compromise users' privacy and determine whether certain users are connected or not in the graph.

All attacks seek to determine whether certain users are connected or not in the graph. It does so by finding each user's node in the anonymized graph, which in turn perfectly tells the attackers which other identified nodes each user is connected to.

This paper demonstrates several effective, practical attacks with different threat models that demonstrate the problems with this simple anonymization technique.

2 Attack Model

We assume the following about the attack model.

First, there is a graph with a node for each account and an edge for each connection. The attacker will see a perfect structural representation of this graph, but nodes will be labeled in an arbitrary manner that conveys no information about who the accounts are.

The full paper also assumes that the graph is undirected, with a connection existing if contact has occurred in either direction (eg. there is an edge between Tim and Daniel if Tim has sent Daniel an email and/or Daniel has sent Tim an email). They state without justification that the undirected case must be harder, but this is plain enough to see; from the directed graph (where separate directed edges would represent Tim sending an email to Daniel and vice versa), we could easily produce the undirected graph. Thus, the directed case cannot be any harder than the undirected case.

The active attack assumes that the attacker is able to create a moderate number of nodes and some edges from those nodes.

While they do not call this out very explicitly, for the active attack, they require what I have termed *non-consensual link creation.* That is, they assume that from accounts we control, we can create any edge that we wish to any other account. This fits if the graph is produced by something asymmetrical such as sending someone an email or following them on Twitter, but not a symmetrical relationship that requires confirmation from the other side such as a Facebook friendship.

In the active attack, they also assume for simplicity that the only edges on the nodes they control are those they create; that is, no one else is creating edges to their nodes. They say that this may not be strictly necessary, and it seems that if we assume that the attacker knows about all edges of which they are a part and those created by others are not too pathological, everything would work just about as well.

The passive attack also assumes full knowledge of the edges the attacking nodes are a part of, despite a lack of control over changing the graph.

3 Walk-Based Active Attack

3.1 Setup and Notation

In the walk-based attack, there is a set W of size b of nodes in the original graph that the attacker wishes to target and deanonymize.

To do so, the attacker will create a set X of k new nodes with some edges between them and some edges to existing nodes in the graph.

The paper calls the subgraph of G that contains only the X nodes. It uses the notation $G[S]$ to be the subgraph of G that only includes the nodes in set S and the edges between pairs of these nodes. It defines $G[X] = H$ and refers to H throughout.

3.2 Simplified Construction

The paper begins by giving a simplified version of the walk-based attack.

In this attack, $k = b$; there are the same number of attack nodes as target nodes.

The construction is very simple. For each pair of nodes in X, (x_i, x_j), we create an edge between x_i and x_j with probability 0.5 independently. Each x_i also has one edge leaving H, to a corresponding w_i.

In the anonymized graph, if the attacker can recover the labelings of the X nodes, they can trivially deanonymize all W nodes; w_i is simply the only node outside of H that is connected to x_i.

3.3 Failure Cases

The paper lists three conditions that are necessary and sufficient for this attack to succeed:

First, **there must be no subgraph of G that is isomorphic to H**, aside from H itself. Failing this, it is impossible to uniquely identify H in the anonymized graph; there would be two or more subgraphs with identical structures in the graph and the attacker would have no way to determine which is H.

Second, **there must be a way to efficiently find H**, given G. Failing this, while the information of which nodes belong to H is encoded in the graph, it would be intractable for the attacker to find H.

Finally, **H must have no non-trivial automorphisms**. Failing this, even if the attacker can identify the set of nodes belong to H, the attacker would be unable to unambiguously label which attack node is which; there would be multiple possible labelings that are indistinguishable from a structural perspective.

These conditions are the same that are necessary for the full attack, and one major goal of the adjustments below is to increase the chance that these hold.

3.4 Potential for Improvement

There are several shortcomings of the simplified construction stated above.

First, there must be one attack node for every targeted node. While this makes identification of the attacked nodes very straightforward and efficient, it requires many more nodes than necessary.

Let us define the subset of X to which target node w_i is connected as S. To uniquely identify w_i, all that is necessary is that w_i be the only node outside of H that is connected to the nodes in S and no other nodes in X. This means that once the attacker has labeled all X nodes in the original graph, w_i will be uniquely identified as the only other node (not already labeled as a member of X) that is connected precisely to S.

In the simplified construction, we fix all sets S to have cardinality of one, yielding a very simple process for mapping from X labelings to W labelings. However, if we allow these sets to be larger, we can attack many more nodes with the same number of attacking nodes.

Another shortcoming is that, while this construction has the benefit that H will have unique structure in G with high probability, there is not obviously a straightforward, reliable, efficient way to find H. While the paper does not explore this, there are also many potential pathological possibilities from this purely random structure, including H not even being a single connected component. Adding some fixed structure to the construction can mitigate this without significantly hurting the uniqueness of H.

3.5 Full Walk-Based Attack

3.5.1 Full parameters

The complete walk-based attack has a few additional parameters and specifies parameters with more complexity.

We choose $k = (2 + \delta) \log n$ attack nodes for a small positive δ and up to $b = O(\log^2 n)$ target nodes.

In addition to these parameters, there are several new ones. The attacker choose $d_o \leq d_1 = O(\log n)$ as the bounds for the *external degree* of each element of X. The attack then requires choosing a $\Delta_i \in [d_0, d_1]$ for each x_i as its external degree–the number of nodes outside of H that it will connect to. The paper recommends selecting uniformly at random from this range, though this is an optimization that the proofs for this attack do not rely on, so values can be chosen arbitrarily.

Another added piece of notation is N_j. Each w_j has a corresponding $N_j \subseteq X$. Each w_j will be connected to the nodes in N_j. Along with this, we add a small constant c that will be the maximum size of any N_j. The paper states that $c = 3$ is sufficient.

Another added parameter is c, the maximum cardinality of the subset of X that any

Δ_i explicitly randomizes the degrees of the nodes in X to produce a wider spread of nodes. This should increase the spread of degrees, allowing the search process to prune nodes more effectively and thus run more efficiently.

3.5.2 Construction

The full construction proceeds as follows:

The attacker creates k new nodes.

For each w_j, we select an N_j and ensure that the global selection maintains:

1. $|N_j| \leq c \; \forall N_j$.
2. All N_j are distinct
3. x_i appears in at most Δ_i sets $\forall x_i$

(1) ensures that c is an upper bound on the size of N_j as intended. (2) ensures that each w_j is uniquely identified by the subset of X to which it is connected. (3) ensures that Δ_i limits to the external degree of x_i's are not exceeded (though it does not ensure that they are met exactly, which is done below).

To create all nodes between H and $G - H$, the attacker does the following:

First, create a link between each w_j and each $x_i \in N_j$.

Finally, to ensure that each x_i has an external degree of exactly Δ_i, for each x_i, we add additional arbitrary links to nodes other than those in W or X. In so doing, we must ensure that each w_j is still the only node in the graph that is connected to precisely the N_j out of the X nodes; if we were to connect some other throwaway external node to N_j in the process of increasing the degrees of certain X nodes, we would not be able to tell this node from w_j when completing the attack.

3.5.3 Search for H

The description of the simplified attack gave no search procedure because, as stated above, there is not a straightforward way to search for the completely-randomized graph it describes. The full attack takes advantage of the known existence of the walk from x_1 to x_2 and so on.

It builds a search tree of possible $x_1, x_2, ...$ paths (in order). Nodes can be filtered by degree and their structural relation to every previous node in the candidate path.

We build the tree downward. The first level contains all candidate x1 nodes, which is simply all nodes with degree 3.

Then, to expand a candidate path, we examine a current leaf node representing some candidate x_i and add a child node below it for any neighbor of that node that is structurally indistinguishable from x_{i+1} for the candidate path we are currently building. This means that it has the same total degree as x_{i+1}, but also that all of its connections to prior nodes in the candidate path correspond to the structure of H.

That is, if we call $\tilde{x}_j$ the candidate x_j in the path we are exploring, then we would add a node $\tilde{x}_{i+1}$ iff it has the same total degree in the graph as the real x_{i+1} and it has a connection to each $\tilde{x}_j$ that precedes it in the candidate path if and only if the actual x_j is connected to the actual x_{i+1}.

Note that the structural requirements gain more power as our candidate path grows longer. While when exploring possible x2's, structural requirements simply allow us to limit our search to neighbors of our candidate x1, when exploring candidate x9's, there are 8 edges (to nodes $\tilde{x}_1$ through $\tilde{x}_8$) that need to exist or not perfectly for us to accept a node as a candidate x9, all but one of which were randomly chosen when H was built. This makes the probability of exploring many false paths fall quickly as the paths grow.

In fact, the proofs rely only on the structural requirements and completely ignore the filtering based on degree. While filtering on degree can drastically improve performance, filtering only based on existence and nonexistence of edges is sufficient to ensure that H is unique and efficiently recoverable with high probability.

An example search tree for the graph in Figure 1 is shown in Figure 2. Each path down the tree is a candidate sequential walk through the x nodes (x1, x2, etc. in order).

An example of pruning based on connections to previous nodes in the candidate path occurs as we add children below the x3 node for the actual path. rg is a neighbor of x3 with a degree of four, the same as x4. However, it is connected the candidate x2 in the path, whereas the real x4 is not, so we do not need to explore rg as a possible fourth node to follow x1, x2, x3.

4 Passive Attack

We have seen the effectiveness of the walk-based attack. H is unique with high probability and has an efficient, straightforward recovery algorithm.

However, there may be cases when it is not possible to construct the walk-based construction before the data is released. Perhaps it is difficult to create dummy accounts, or perhaps the anonymized graph was released before the attacker had the foresight to manipulate the graph.

The paper describes a passive attack for this case that does not require manipulating the graph whatsoever. Instead, a coalition of connected users leverage the fact that, while they cannot adjust they graph, they do have significant combined knowledge about the graph.

Figure 1: Attack graph in which search in 2 takes place. Green attack nodes (x1 through x5) make up X, and all nodes with some blue connection to them are targeted and uniquely identifiable from the anonymized graph.

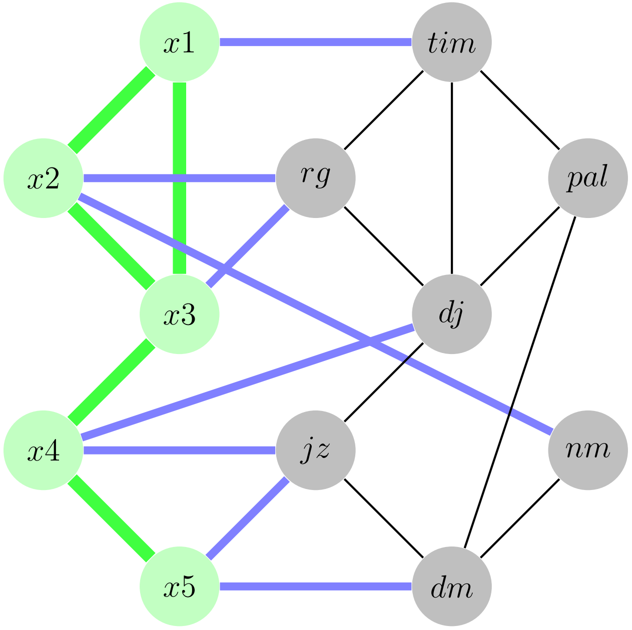
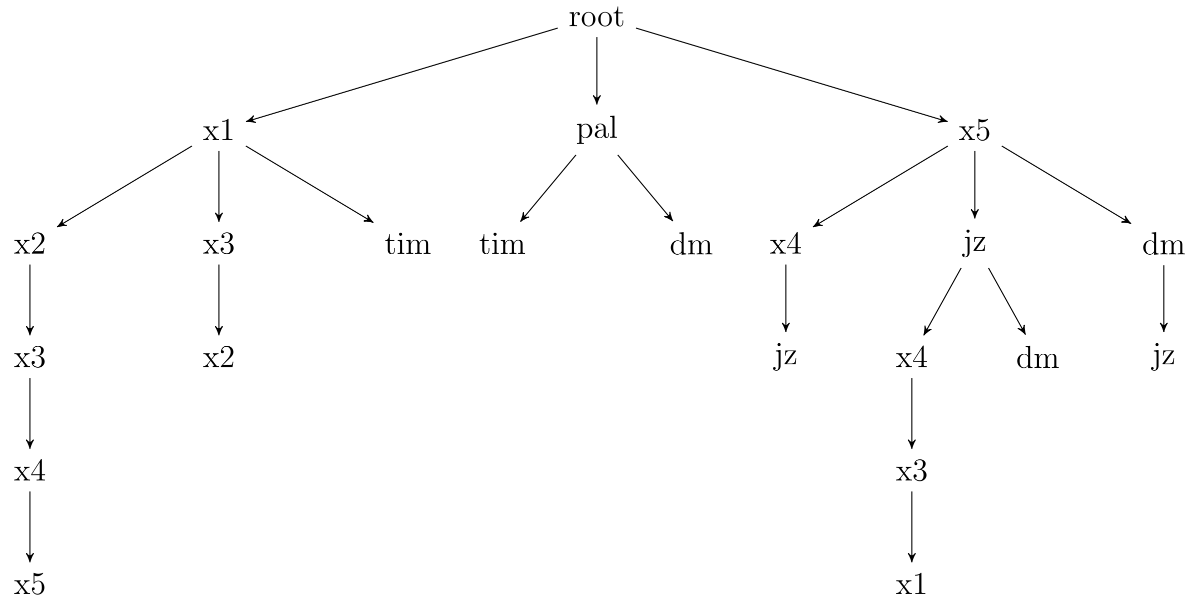


Figure 2: Example of searching for H in Figure 1. The attack will fail in this case, as the attacker will be unable to differentiate the real path from one that is structurally identical.



If they are able to use this to find themselves in the anonymized graph, they can likely compromise the privacy of some users to which they are connected.

The process is the same as in the walk-based attack–but without being assured a simple walk between them. As the paper describes the attack, it assumes that all users are connected to one central user who assembles the coalition.

The details of the modifications are elided, but presumably an arbitrary ordering is given to the nodes in the coalition and the attack again searches one at a time for those nodes, pruning any path that is not properly connected to the previous candidates.

If the coalition is found in the graph, then the attackers can identify any nodes that happen to be connected to a unique subset of the coalition.

One advantage of the passive attack that goes unstated in the paper is that it works even in networks that do not have the *non-consensual link creation* property. Since the attack does not require creating any new edges, whether or not this property applies to the network in which the attack occurs is irrelevant to the attack.

4.1 Semi-Passive Attack

The paper also describes a semi-passive attack for models where the attacker can create edges but not accounts.

In this attack, the coalition first gathers their information about who they are connected to and then identifies subsets of the coalition that are currently not precisely connected to any node in the graph. To target a user, the coalition need simply choose one of these unallocated subsets and have each coalition member in that subset create an edge to the targeted user. This user is now the unique node in the graph connected to this subset, and can thus be identified by the passive attack.

The semi-passive attack still allows targeting attacker-chosen users, so it demonstrates that being unable to create new accounts is not a major barrier to this class of attack. The main drawback is that recovery is somewhat less likely and likely less efficient because the coalition's structure has not been engineered to have a high probability of uniqueness and a straightforward recovery process.

5 Experiments

They perform several experiments to evaluate the actual effectiveness of their algorithms and the effect of parameters.

All experiments are done in a 4.4-million node LiveJournal graph.

In the first experiment, they examine how the number of attack nodes and degree range of those nodes affects recoverability of H. The produced graph is below:

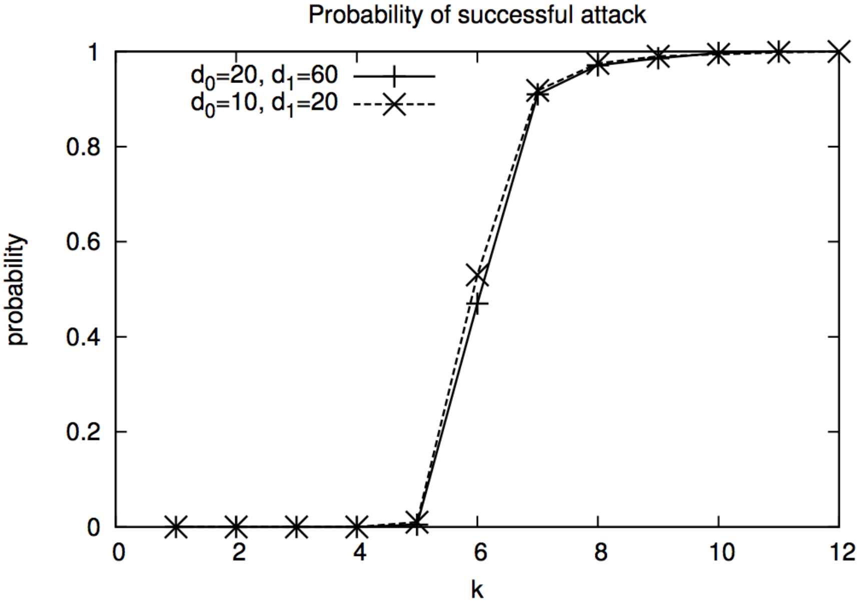


Figure 1: For two different choices of d_0 and d_1, the value $k = 7$ gives the attack on the LiveJournal graph a high probability of success. Both of these choices for d_0 and d_1 fall well within the degrees typically found in G.

Seven attack nodes yields a high probability (around 90%) of recovery, and the external degree matters little for the two ranges they attempt.

Next, they examine the recoverability of the passive attack:

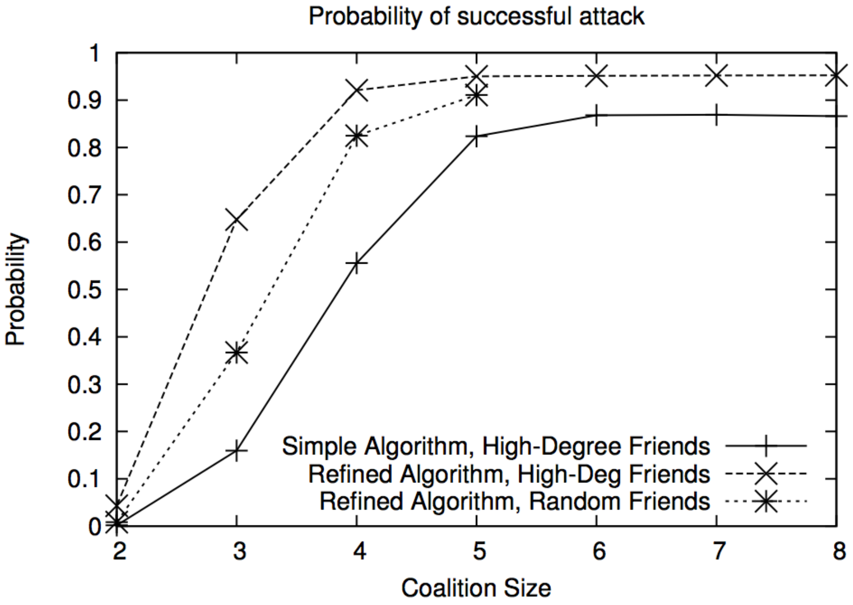


Figure 2: Probability of success for different coalition sizes, in the LiveJournal graph. When only the degrees and internal structure of the coalition are taken into account, a coalition of size 5 is needed to give a high probability of success. When the more refined version of the algorithm is used, and the edges connecting H to $G - H$ are considered, only 4 users need collude.

With one added optimization, it is competitive with the walk-based attack for this graph. Finally, they explore how many targets the passive and semi-passive attacks can exploit:

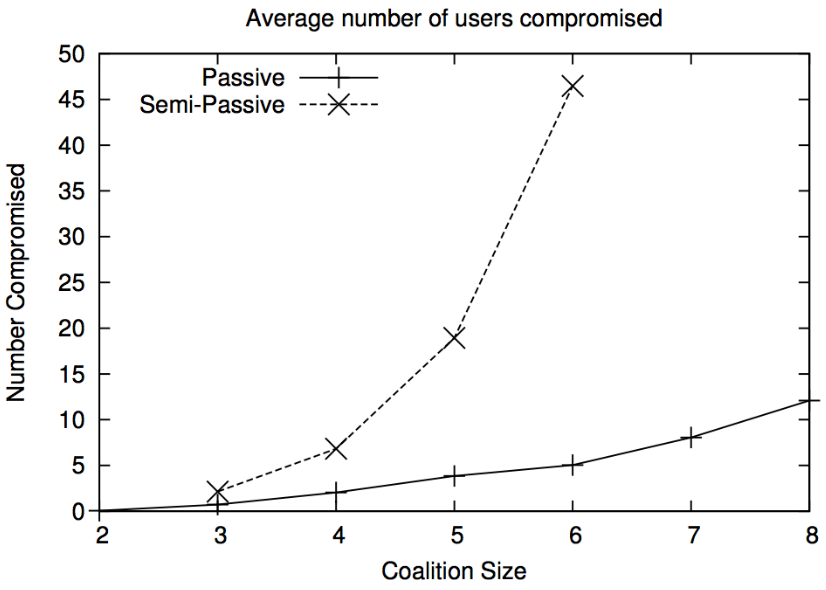


Figure 3: As the size of the coalition increases, the number of users in the LiveJournal graph compromised under the passive attack when the coalition successfully finds itself increases superlinearly. The number of users the semi-passive attack compromises increases exponentially.

In the passive attack, the number is fairly low and slightly better than linear. This makes sense since they must get lucky and have a unique subset of themselves connected to each exploited user.

In the semi-passive attack, the number exponentially increases because it is a function of the total number of subsets of the coalition.

These experiments show that for modest numbers of nodes, these attacks are quite effective.

6 Proof of Theorem 2.1

We will now walk through much of the proof that in the walk-based attack, H can be recovered with high probability.

The goal is to put an upper bound on the probability that some other subgraph of G will be isomorphic to H, and show that this bound exponentially approaches 0 as k grows to infinity, in turn proving that H will have a unique structure with high probability.

The proof has a few high-level steps:

6.1 Claim 1

It starts by defining $\mathcal{F}_0$, the event that there is no subset of nodes S disjoint from X such that $G[S]$ is isomorphic to H. The difference between this event and the event we are ultimately trying to bound is that it requires S be disjoint from X, ie. have no overlap with X. However, if there is any S that simply differs from X at all (even if some nodes are in common), then H will not be uniquely recoverable. Thus this is a weaker claim than what we ultimately need to show that H will have unique structure in the graph with high probability.

They prove that $\mathcal{F}_0$ holds with high probability.

They define ε_S as the event that for the ordered sequence $S = (s_1, ..., s_k)$, the function $f(s_i) = x_i$ is an isomorphism from S to X.

There are two keys to easily bounding $Pr[\varepsilon_S]$. First, S must be disjoint from X. Second, all but $k-1$ edges in H are chosen with probability 0.5 IID. The combination means that we can view this event as requiring the many coinflips made in H to perfectly align with the outcomes set by the structure of $G[S]$.

In particular, there are $\binom{k}{2}$ edges, the existence or nonexistence of all of which must that must all be the same in H and $G[S]$. Only the $k-1$ edges in the path $x_1, x_2, ..., x_k$ are not simply coin flips.

Thus, $Pr[\varepsilon_S] = 2^{-\binom{k}{2}+(k-1)} = 2^{-\binom{k-1}{2}}$.

$\mathcal{F}_0$ is simply the union over all S's disjoint from X in the graph. There are strictly fewer than n^k of these. Thus, by the union bound, if ε is the complement of $\mathcal{F}$, then

$$Pr[\varepsilon] < n^k 2^{-\binom{k-1}{2}}$$

which they manipulate to show that it goes to 0 exponentially quickly in k.

6.2 Claim 2

Claim 2 is similar to claim 1, so its proof is omitted from the paper, but it is needed by claim 3.

It defines the $\mathcal{F}_1$ as the event that, for a constant $c_1 > 4$, there are no disjoint sets of nodes Y and Z in H, each of size $c_1 \log k$, such that H[Y] and H[Z] are isomorphic.

Claim 2 is that $\mathcal{F}_1$ holds with high probability.

6.3 Claim 3

The high-level goal of claim 3 (as with claim 2) is to argue that there is not much internal symmetry in H; this claim, combined with the low probability of other parts of the graph having the same structure as H, can prove that H is likely to be unique.

We do this by bounding certain quantities of isomorphisms involving nodes form H. One of the bounds limits the number of nodes involved in an isomorphism that are not fixed points. A fixed point of an isomorphism f is a node s such that $f(s) = s$. Intuitively, if we show that any isomorphism of H must mostly be fixed points, there is little internal symmetry; there are not many parts of H that are symmetrical to other parts of H.

Claim 3 supposes the following: $\mathcal{F}_1$ holds. $c_2 \geq 3c_1$. We have three disjoint sets of nodes in G: A, B, and Y. $B, Y \subseteq X$ are subsets of the attack nodes. $f : A \cup Y \to B \cup Y$ is an isomorphism; that is f, is an isomorphism mapping from A and Y to B and Y.

It claims that given these conditions, a) $f(A)$ contains at most $c_1 \log k$ nodes not in B, and b) Y contains at most $c_2 \log k$ nodes that are not fixed points of f.

Both parts of this proof put limits on what kinds of isomorphisms can exist involving H.

The proof of both results are argues using a simple directed graph, K. The nodes of K are $A \cup B \cup Y$ (ie. the union of the domain and range of f), and there is a directed edge from v to w iff $f(v) = w$.

Since f is an isomorphism from $A \cup Y$ to $B \cup Y$, we know much about the structure of this graph. Nodes in A have out-degree 1 and in-degree 0 (they can only map to one node since f is a function, and no node can map to them because A is not in the range of f). Nodes in B are the opposite, with out-degree 0 (they are not in the domain of f) and in-degree 1 (isomorphisms are bijections, so only one node can map to each node). Nodes in Y will have an in-degree and out-degree of 1 because they are in both the domain and range of f.

This means that there are only two kinds of paths through the graph. Cycles made up entirely of Y nodes (including self-loops, which are simply cycles of size 1); and paths that start in an A node, traverse some (possibly trivial) number of Y nodes, and then end at a node in B. They define a non-trivial path component to be a path from A to B (the latter kind of path in the graph) that includes at least one node from Y.

Figure?

Claim a) falls out of the claim that there can be at most $c_1 \log k$ nontrivial path components in K. This is proved (implicitly) by contradiction. Suppose there are more than $c_1 \log k$ nontrivial path components in K. Let $Y' \subseteq Y$ be all penultimate nodes from each path (the last Y node, each of which will point to a B node). We know that $f(Y') = B'$ for some $B' \subseteq B$.

Since there is an edge from each node in Y' to B' in K, and edges represent isomorphism in G, $H[Y']$ and $H[B']$ are isomorphic. Y' and B' are subsets of Y and B respectively, which are disjoint subsets of X. Thus, they are disjoint subsets of X. Since we have assumed that there more than $c_1 \log k$ in each set ($|Y'| = |B'|$, we have two disjoint subsets of size more than $c_1 \log k$. Thus, we have contradicted claim 2. Since assuming that there are more than $c_1 \log k$ nontrivial path components in K has produced a contradiction, we have proven that there are not more than $c_1 \log k$ nontrivial path components in K.

Claim b) again uses the graph. To bound the number of elements of Y that are not fixed points of f, we first define Z to be the set of nodes in Y that are not fixed points of f. To bound the size of this set, we select every other edge on each cycle and path, starting with the second edge on each path.

This yields at least $|Z|/3$ edges; a length-3 cycle that gives us one edge is the worst case. We define $Z_1 \subseteq Z$ as the tails of the selected edges and $Z_2 \subseteq Z \cup B$ as their heads. This means that $f(Z_1) = Z_2$, and thus that $G[Z_1]$ and $G[Z_2]$ are isomorphic.

Z_1 and Z_2 are disjoint subsets of X; since we chose every other edge on a path, and each node has one edge coming in and one edge coming out, no node will have more than one edge it touches selected.

Since $\mathcal{F}_1$ holds by assumption, we know that $|Z_1| = |Z_1| \leq c_1 \log k$.

Since $|Z_1| \geq |Z|/3$, this means that $|Z| \leq 3c_1 \log k \leq c_2 \log k$, proving part b) of Claim 3.

6.4 Completing the proof

With these claims proven, the remainder of the proof is simply using them and carefully bookkeeping the number of possible subsets of G of different sorts to show that the probability of any isomorphism to H goes to 0 as n increases.

7 Conclusion

These algorithms show that simply removing identifiers from a social graph is an ineffective anonymization mechanism that will not preserve users' privacy reliably. It demonstrates several effective attacks, even if the attackers cannot create more accounts and even more edges.

They are shown to work with high probability in practice with modest needs for adding to the graph, and some success even without adjusting the graph. They also incorporate smart optimizations that are unneeded by theory, but that the authors claim yield significant performance wins.

While research in social networks is important, if untrusted researchers need access to data, a more sophisticated mechanism is necessary to avoid violating privacy.