Progress Report: Deanonymizing Anonymous Messaging Social Networks

Giulia Fanti and Wenting Zheng

Sketch of project: Our project is investigating the anonymity properties of commercial anonymous social networks, like Secret, Whisper, and Yik Yak. These networks allow people to spread messages over a contact network without authorship information attached. The spreading mechanism occurs as follows: each time a user approves a message (e.g., retweeting on Twitter), that message becomes visible to the users neighbors on the social graph. We are considering an adversary that consists of colluding spy nodes in a social network. This is meant to simulate a government agency that recruits agents to participate in a social network in order to monitor the content that flows through it. Our hypothesis is that such an adversary can infer the true source of a message simply by measuring which spy nodes observe the message, and the time at which messages are received.

To test our hypothesis, we are simulating the spread of messages over various networks in the presence of spy nodes. Given a simulated message spread, we try to infer the true source from the spy nodes information. We vary attributes such as graph topology and size, the number and distribution of spy nodes, the probability of a node approving a message, and the estimation algorithms used. Our goal is to understand how many spies an adversary must corrupt in order to reliably infer the source of a message, regardless of where in the network it originates.

Work completed: Thus far, we have written a simulator for spreading messages over a connectivity network. This simulator can generate synthetic graphs (Barabasi-Albert, Watts-Strogatz), and it can also use existing graph datasets. We have found a Facebook dataset that we plan to use in our simulation, which contains the Facebook links among about 10,000 nodes in 2009. Given a graph, the simulator allows nodes to make decisions about when and how they will spread a message to their neighbors. So far, we have decided to model the whole system as a discrete-time system. Nodes decide to approve messages by drawing samples from a Bernoulli random variable with parameter 0.5; we have modeled the delay with which a node sees a transmitted message as a geometric random variable with parameter p.

We have also implemented a Jordan centrality estimator, which guesses that the true message source is the node with the lowest Jordan centrality. The Jordan centrality of a node is defined as the distance from that node to the farthest node on the graph. In our case, we are finding the node with the minimum distance to the farthest-away spy node in the network. This is the most basic estimator we will test, as it does not even use timing information from the spies.

We have tested this existing code on a 300-node Barabasi-Albert graph.

Additionally, we have changed our project definition slightly after realizing that inserting Sybils into the network is equivalent to saying that the legitimate end of the attack edges are spies. Therefore, Sybils do not give any additional information; as such, we can understand the threat that Sybils pose by understanding the proportion of network nodes that befriend Sybils, and treating them as spies.

What remains: At this point, we need to run a lot of different simulations in order to explore the parameter space adequately. We will also consider more complex estimators, but these are likely to be heuristics, since optimal estimators in this space are difficult to compute. It seems like the path forward will be fairly methodical.

We are also planning to test these estimators when the adversary knows only a subset of the underlying graph. This is a more likely scenario in practice, because it might be difficult for an adversary to learn the whole graph if the social network is maintained by a private company. Of course, this depends on the strength of the adversary. For this sampled-graph scenario, we may need to develop new heuristic methods for inferring the true message source over partial estimates of the underlying network. Then we will need to run the simulations from the full-graph scenario on the partial graph.

Concerns/open issues: Our main concerns at this point are that the computational load of these simulations will become too prohibitive to explore the parameter space fully. We can try to optimize the code somewhat in this regard, but parallelizing graph algorithms may be difficult.

Need for meeting/availability: We dont think we need a meeting (unless you feel otherwise).

Presentation slot preference: We would prefer to present on May 6 or 8, as Giulia has to proctor a midterm on May 1 and may need to proctor during our usual class slot.