Improving Anonymity using Social Links

Krishna P. N. Puttaswamy, Alessandra Sala and Ben Y. Zhao
Computer Science Department, U. C. Santa Barbara
{krishnap, alessandra, ravenben}@cs.ucsb.edu

***Abstract*—Protecting user privacy in network communication is vital in today's open networking environment. Current anonymous routing protocols provide anonymity by forwarding traffic through a static path of randomly selected relay nodes. In practice, however, malicious relays can perform passive logging attacks to compromise the anonymity of a flow. This degradation is accelerated when nodes fail, forcing source node to reconstruct a path, and in doing so, leaking more information to passive loggers. This "predecessor attack" is highly effective and difficult to defend against on current systems. In this paper, we propose a highly effective approach to blocking predecessor attacks by leveraging trusted links from social networks. We first show how users can completely shield themselves from traditional logging attacks. We then propose a hybrid logging attack optimized for social networks, and perform detailed analysis to show that we can defend against it using optimized path selection techniques. Finally, we analyze detailed measurement traces from Facebook to show that our approach is indeed feasible given the user behavior in social networks today.**

I. Introduction

Protecting user privacy in network communication is vital in today's open networking environment. More specifically, many applications desire the ability to hide the identity of the communicating parties from each other and third-party observers. Anonymous routing is used in many applications such as anonymous web browsing [2] and anonymous voting.

While a number of anonymous communication protocols have been proposed [5], [6], most of them are prone to passive logging attacks like the *predecessor attack* [21], [10]. In this attack, attackers log the participants in an anonymous path. As nodes fail or exit from the network during a session, paths must be rebuilt. Attackers then correlate observed participants over multiple paths to identify the communication endpoints, since they must participate in each rebuilt path. This has been shown to be highly effective in both theoretical analysis and practical on the popular Tor network [3]. Note that this threat is more severe for longer communication sessions with a higher number of path rebuilds. As an increasing number of Internet applications move towards a web services model, we expect anonymous sessions to grow in length, further exposing users to this type of attack.

Current defenses against these attacks are limited and ineffective. One approach is to leverage "persistent nodes" in the path to shield the end-points and limit their exposure to attackers [20]. This was adopted into the Tor network [5] as "guard" nodes. However, this solution is difficult to realize in practice, since persistent nodes are rare in real systems due to the complexity and costs of maintenance. A recent Tor measurement study showed that very few nodes in Tor have the stability and resources to serve as guard nodes [11]. In addition, the few nodes that are able to serve immediately become high-yield targets for attacks. Another recent study showed that attacking only a few nodes could compromise the anonymity of a significant portion of the Tor network [3].

In this paper, we describe and investigate a highly effective defense against logging attacks leveraging the prevalence of trusted social links between online users. Recent years has seen rapid growth in social networks such as MySpace (190+ Million users) and Facebook (80+ Million users). Measurements show that the average social network user has anywhere between 5-150 direct friends [12]. We propose that anonymous networks be built to leverage these social networks: users would join an anonymous network along with their friends. Instead of centralized infrastructure-based guard nodes, participants can use trusted friend nodes to shield them from passive logging attacks. Not only do these social links protect the source node from malicious loggers, but their distributed nature means that load is spread across the network, thus avoiding tempting targets to attack and limiting the loss of anonymity following a successful attack.

This paper makes three key contributions. First, we propose and evaluate the effectiveness of several algorithms for using social links to build buffers against logging attacks. Users must utilize their social network wisely to defend against attackers with knowledge of the social graph topology. Second, we describe a novel two-phased attack against social anonymous networks that reduces sender anonymity. We perform detailed analysis to quantify its effectiveness, and use our results to derive a more attack-resistant path construction algorithm based on cliques. Finally, we study the feasibility of our approach using measurement traces of Facebook, Tor and Gnutella. We show that most social network users have sufficient number of online friends to protect them across lengthy sessions, and most users belong to sufficiently large cliques to ensure strong anonymity against even our two-phased attack.

The rest of this paper is structured as follows. We describe related work in Section II, and then we describe our assumptions and proposed design in Section III. Next, we propose a two-phased logging attack, and analyze its impact on path construction algorithms in Section IV. We then use measurement results to demonstrate the feasibility of our approach in Section V, and finally conclude.

II. Context and Related Work

We describe our work in the context of well-known Onion Routing [5], [17] protocol, which has been thoroughly analyzed before [17], [21]. However, other protocols such as peer-to-peer anonymous routing [6], [23] and mix networks [4] can benefit from our work in the same way.

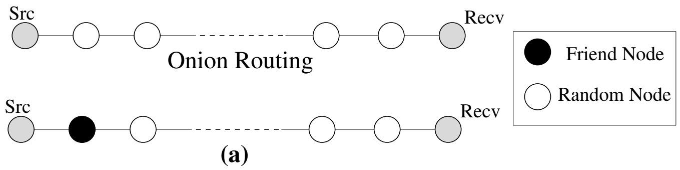

Fig. 1. Onion Routing path consists of randomly selected nodes. Our proposal, (a), has a friend node in the first position, with rest of the path members chosen randomly.

In Onion Routing, the source initiates a low-latency anonymous communication with a receiver by setting up a path consisting of randomly selected nodes in the network. The path starts with the source and terminates at the receiver as shown in Figure 1. We assume that the source and the receiver are not malicious, but the randomly chosen intermediate nodes on the path can be malicious attackers.

Since our goal is to improve anonymity through the addition of *trusted* nodes in the path, we focus mainly on the node selection component of path building. Unless otherwise specified, we utilize the well-studied Onion Routing path setup algorithm for other components [5].

To simplify our analysis, we consider the problem of securing a single flow against attacks. While there might be multiple flows at any given time between different pairs of endpoints, we assume that attackers can disambiguate between different flows using timing correlation.

Finally, social networks have been the focus of several recent measurement studies [12], [1]. Following their integration into file-sharing networks [18] and online auction systems [16], the research community has also proposed to exploit trust from social networks to improve the security of email and distributed systems [7], [22], [13].

III. System Model and Design

We describe our system model, followed by an overview of our design and detailed strategies for path construction.

A. Assumptions and Attack Model

First we define terminology used in this paper. We refer to each participant in the network as a *node*, the node initiating an anonymous connection as the *source*, and the destination of the anonymous connection as the *receiver*. An anonymous *path* consists of a number of intermediate proxies each referred to as a *relay*. Continuous communication between a source and a receiver is called a "session." As node churn disrupts sessions, paths are rebuilt to reconnect the two endpoints. We refer to the time period between two successive rebuilds of a path as a "round."

We assume that nodes in the anonymous network belong to a social network, and have a set of friends from that network who are also participants in the anonymous network. For example, we can deploy a peer-to-peer anonymous network on top of the Facebook application platform. Note that users in the anonymous network do not need to belong to the same social network S, as long as their friends from outside S also participate in the same anonymous network. We also assume that a user has access to both her list of friends and also their lists of friends, as is the case on Facebook. Moreover, we design a strategy to discover nodes in the k-hop social neighborhood, so that each source is able to build a stronger anonymous path prefaced by hops in this social neighborhood. We describe the discovery procedure in detail, and formally prove our anonymity benefits in Section IV.

Attack Model. Like prior work on passive logging [21], we assume a model where attackers are users who can passively monitor and log their communication with other nodes. We assume that in a network of N nodes, c (where $c < N$) users are malicious attackers who can collude with each other and share their logs with zero delay. However, attackers do not perform active attacks such as dropping or corrupting packets. We further assume that the attackers can perform timing attacks and hence can successfully perform predecessor attacks [21].

In addition to performing the predecessor attack, we assume that the attackers are powerful enough to obtain the full social network topology, *e.g.* by crawling the social network. Given this information, a group of attackers can tailor customized logging attacks for social network enhanced anonymous systems. We describe such an attack later in Section IV. Finally, we also allow the possibility that an attacker can compromise any node in the network, and can therefore passively observe the traffic routed by the local user for her friends.

Note that this is a highly conservative model. In practice, attackers will find it difficult to obtain complete connectivity data for a social network, and is highly unlikely to compromise all friend nodes of an anonymous user.

B. Anonymity via Social Networks

At a high level, we are proposing that source nodes in anonymous systems leverage the presence of "trusted" friend nodes to shield them from being observed by passive attackers. We rely on an assumption of inherent trust between links in the social network. This trust comes from both social relationships established between users in the real world and from explicit authorization required to become "friends" in a social network. Social links have been shown to be effective in introducing trust into a variety of applications [18], [16]. More specifically, a recent measurement study of a social auction system confirmed that even transitive social links help protect users from malicious attackers by filtering them from the social network [16].

Friend Selection. Each user has access to a list of her friends and each friend's list of friends (friends-of-friends, or FoF). Using this information, the source node can construct a communication path such that trusted friends or FoFs are inserted into the path to block attackers from observing the source or destination.

Using only one-hop friends as shields is risky. With the social link topology, an attacker can use a successful predecessor attack to narrow down the source to a smaller set of users. Instead, we propose that the source choose k nodes from its k-hop social "neighborhood." This consists of all nodes reachable within k social hops from the source. We refer to this neighborhood as the friend-of-friend (FoF) network. We show later via experiments that we achieve sufficient neighborhood sizes using k = 2.

Path Formation Strategy. The source node can utilize its FoF nodes in different ways to improve anonymity. The FoF nodes can be positioned as the first node on the path to protect the source, as the last node on the path to protect the receiver, in both positions, or at all positions along the path. We focus on using FoF as the first hop to improve source anonymity, and defer analysis of other strategies for future work.

In this strategy, the source s constructs the path with a trusted FoF node F as the first relay after the source. Other relays in the path are chosen at random. Unless an attacker compromises the FoF, no attacker can observe the source directly. From the attackers' point of view, F could either be a random relay, the source node, or a friend of the source node. If F fails or leaves, the source reforms a new path replacing F with a new friend. However, if a non-FoF relay leaves, the source rebuilds the path using the same FoF node F, replacing only the randomly chosen nodes. This way of reusing FoF nodes exposes fewer FoF nodes to the attackers and hence limits information leakage.

Path Construction with FoF. First, a source must obtain a set of nodes within its $k-$hop social neighborhood. We propose that all nodes send periodic neighbor discovery messages to its friend nodes in the anonymous network. Each node appends its own nodeID to the message before forwarding it on, until a per-message time-to-live parameter expires. This background discovery traffic periodically updates each user with a list of currently online friends[1]. Alternatively, a user can query a friend for a list of their online friends. When a node discovers new FoF nodes, it queries for their public key from a Certificate Authority, and caches the key.

To construct a path, the source chooses at random a sequence of k socially-connected nodes in its FoF neighborhood. It uses the sequence of FoF nodes as the preamble to the path, and chooses random relays to form the rest of the path. While there might be up to k FoF nodes on the path, we treat the entire FoF chain as one friend for the sake of analysis.

Note that a compromised friend in this FoF chain does not gain any more information than a random malicious node following the chain. Indeed, a compromised FoF does not know its location on the chain, and therefore cannot determine if its predecessor is the source. We treat a compromised FoF the same as a malicious node, and the following analysis applies also in this scenario.

IV. Analysis of Attacks and Defenses

In this section, we analyze the resilience of our approach against passive logging attacks. First we describe why our approach is resilient against both the predecessor and intersection attacks. We then present a new two-phase passive logging attack, and use detailed analysis to show its impact on our system. Finally, we describe a modified path construction algorithm that provides stronger protection against the two-phase attack.

[1]While this could provide limited topology information to potential attackers, we already assume that attackers can crawl the full social network.

A. A Two-Phase Attack

Two logging attacks have been proven most effective against anonymous systems: the predecessor attack which passively logs traffic across rebuilt paths, and the intersection attack, which combining sets of nodes that possibly contain the communicating end-points. However, neither of these attacks, as described by the literature [19], [20], is effective against our approach. The predecessor attack fails, because the source always hides behind one or more FoF nodes, and is never observed directly by an attacker. The best the predecessor attack can do is to identify the FoF node associated with a particular flow. On the other hand, the intersection attack is ineffective in our attack model, because attackers do not have global knowledge of the online/offline status of nodes in the network.

The attackers can, however, combine a modified version of the predecessor attack with a version of intersection attack to attack our system. In this combined attack, the attackers first try to narrow down the possible friends of friends used to shield the source. After the FoF nodes have been sufficiently logged, the attackers can perform an intersection attack using the social link structure information they have to identify the possible source. This is what we call as *two-phase* attack.

In *Phase I*, the attackers perform a modified predecessor attack. Each time an attacker receives a flow from a node x, the attacker stores the following information:

- the identity of the node x from which it got the flow;
- all nodes within $k-$hop social distance of x, because the source can be any node between 1 and k hops. We represent the friends at distance 1 as:
  $F_{x,1} = \bigcup_i f_{x,1,i} = \{f_{x,1,1}, f_{x,1,2}, f_{x,1,3}, ...\}$;
  the friends at distance 2 as:
  $F_{x,2} = \bigcup_i f_{x,2,i} = \{f_{x,2,1}, f_{x,2,2}, f_{x,2,3}, ...\}$;
  and so on until the friends at distance k:
  $F_{x,k} = \bigcup_i f_{x,k,i} = \{f_{x,k,1}, f_{x,k,2}, f_{x,k,3}, ...\}$
  We aggregate the previous sets of nodes in order to have a more compact notation and avoid duplicate nodes as: $F_x = \bigcup_{i=1}^{k} F_{x,i}$.

Phase II begins when the attackers find the possible sets that include the source with high confidence, and proceeds:

- Select the most seen sets among F_x, for any observed x
- $\bigcap_{for\ any\ x} F_x = \{v | v \in F_x\ for\ any\ x\}$

The attackers should narrow down in Phase I the possible friend sets (F_x for any x at distance $\leq k$ from the source) by observing the flows. Sufficient number of observations must be made in order to identify with high probability which logged friend sets contain the real source node. Then in Phase II, the attackers use a threshold of observations to filter out which observed friend sets contain the real source, and perform an intersection across all of them to isolate the identity of the source. If Phase II begins too early, i.e. with an insufficient number of observations, then an incorrect friend set, one that does not contain the real source, can be included in the set intersection. This intersection then proceeds to remove the true source from the result set, and guarantees an incorrect result.

Performed correctly, the attackers can identify after Phase II the source of the communication. However, compared to existing anonymous routing proposals [5], [19], our system requires significantly more number of observations by the attackers (and therefore more time). In the next section, we present detailed proofs to quantify the necessary number of rounds observed (recall that a round is the time between successive rebuilds of a path). These results serve to quantify the robustness of our social-based defenses, and also determine the necessary threshold that attackers must attain before proceeding to Phase II.

B. Analytical Results

First, we need to understand the number of rounds the attackers should spend in the worst case before moving to the second phase. In Theorem 1, we bound the number of rounds needed in Phase I of the attack to identify, with high probability, the source's $k-$hop neighborhood.

Theorem 1. *The number of rounds that "c" colluding attackers have to perform passive logging in the network is* $O((\frac{N}{c})^2 f \log N)$.

The proof follows guidelines proposed in [21]. For space considerations, we omit the proof and interested readers to our technical report [9].

At this point, the attackers are ready to proceed to Phase II of the attack. The attackers only need to intersect the most logged sets of nodes hoping that the intersection will result in a small number of nodes that includes the source node.

Theorem 2. *At the end of Phase II, the attackers may identify the source node.*

Proof: During Phase I, f distinct friends have been recognized. As proven in Theorem 1, the attackers have to log nodes for $T > 8(\frac{N}{c})^2 f \log N$ rounds in order to observe the source's k-hop social neighborhood more than $\frac{1}{2}(\frac{c}{N})^2 \frac{T}{f}$ times, and therefore be sure that they are considering the f friends of the right source. If Phase I ends too early, *i.e.* with an insufficient number of observations, then an incorrect friend, one that is not in real source's k-hop social neighborhood, can be included among the friends used in Phase II. The intersection in Phase II then proceeds to remove the true source from the result set, and guarantees an incorrect result. Therefore, the attackers must wait $T > 8(\frac{N}{c})^2 f \log N$ rounds before starting the intersection Phase. The attackers have already stored the nodes in the k-hop neighborhoods of each one of these f nodes. Let $F_{x_1}, F_{x_2}, ...F_{x_f}$ be the most logged sets during the first phase (T rounds) of the attack such that $x_1, x_2, ..., x_f$ are the f distinct friends in k-hop neighborhood of the real source. Then, if there exists a combination of a subset of $1 \leq i \leq f$ of $F_{x_1}, F_{x_2}, ...F_{x_f}$ such that $\bigcap_{j=1}^{i} F_{x_j} - \{source\} = \emptyset$, then at the end of the intersection phase, the attackers can recognize the real source. This is possible because a combination of subsets could exist whose intersection consists of only one node, the source. This comes from our empirical studies on the Facebook network, where most nodes belong to multiple distinct cliques that only intersect at the source nodes. ■

Optimization Based on Clique Selection. While our approach forces attackers performing the two-phase attack to pay a high cost in rounds, it can ultimately succumb to a group of persistent attackers. We now describe an optimization to our path construction algorithm that prevents the attack from recognizing the source node, but only allows attackers to identify a clique that includes the source. The optimization uses cliques as shields rather than individual FoF nodes.

Intuitively, members of social networks with similar "interests" are tied in together in the social graph. This process leads groups of friends to form maximally connected cliques. In fact, an entire social network can be viewed as cliques of friends connected via common friends. Each node in the social network may be a part of multiple different cliques, *e.g.* a clique of colleagues, a clique of students in the same class, etc. We use detailed measurements to justify the prevalence of cliques in social networks in Section V.

In this modified algorithm, we exploit these cliques in path construction. The source uses one of its friends in its biggest clique as first hop of its $j-$hop random walk, with $1 \leq j \leq k$. As before, the random walk proceeds to include $j-1$ friends until j hops are exhausted. By using this strategy to include cliques right after the source, we obtain a stronger result on source anonymity, as shown in Theorem 3.

Theorem 3. *Phase II of the two-phase attack ends when it identifies all the members of the source node's biggest social clique, with each member equally likely to be the source.*

Proof: Let's again define $F_{x_1}, F_{x_2}, ...F_{x_f}$ for any x_i as the sets of the most logged nodes during the first phase ($T > 8(\frac{N}{c})^2 f \log N$ rounds) of the attack. Because of the fact that each of the x_i nodes are at most $(k-1)$-hop away from one of the source's clique nodes, all the source's nodes in its biggest clique will appear in each of F_{x_i} sets. The nodes that belong in the source's biggest clique are always used as the first hops of the FoF chain. Therefore, on the reverse path from the attacker to the real source, all of these nodes are indistinguishable from the real source. Therefore, $\bigcap_{i=1}^{f} F_{x_i} = \{v | v \in source's\ biggest\ clique\}$. ■

To summarize, Phase I of our attack proceeds asymptotically f times slower than the predecessor attack [21]. In addition, we showed that we could improve our defense against Phase II of the attack by modifying our path construction to utilize cliques. This prevents source nodes from being identified, further improving source anonymity.

V. Feasibility Study

In this section, we evaluate the feasibility of our proposal using simulations driven by measurements of Tor, Facebook, and Gnutella. We first try to understand the size of friend-of-friend networks for real social network users. Larger FoF networks means more online friends that can shield the user from attacks. To this end, we obtained an anonymized dataset

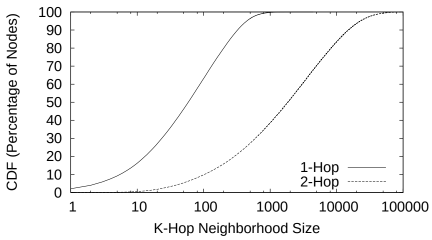

Fig. 2. CDF of neighborhood size in Facebook for different neighborhood depths.

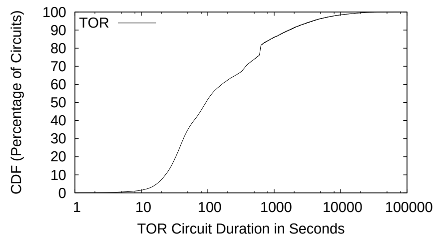

Fig. 3. CDF of TOR circuit duration from our TOR measurements.

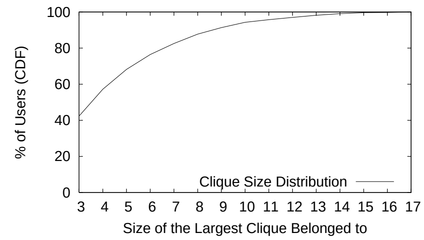

Fig. 4. CDF of users and their largest clique sizes in the Monterey Bay network.

Friend-of-Friend Neighborhood Size	1000	2000	3000	4000	5000
Live Neighbors in Tor (% live after 20 mins)	528 (87.50%)	1083 (87.41%)	1635 (88.41%)	2178 (88.53%)	2639 (88.10%)
Live Neighbors in Gnutella (% live after 20 mins)	275 (91.50%)	593 (91.12%)	850 (91.4%)	1168 (90.92%)	1479 (91.34%)

TABLE I

FOR USERS WITH FRIEND-OF-FRIEND NETWORKS RANGING IN SIZE FROM 1000 TO 5000, WE SHOW THE NUMBER OF EXPECTED LIVE NODES IF USER AVAILABILITY FOLLOWS MEASUREMENTS OF THE TOR NETWORK OR THE GNUTELLA NETWORK. VALUES IN PARENTHESIS SHOW THE PERCENTAGE OF THOSE LIVE NODES THAT WOULD REMAIN ONLINE THROUGHOUT A 20-MIN SESSION.

of 380,000 user profiles from Facebook's New York City regional network[2].

K-hop Neighborhoods in a Friend-of-Friend Network. We define the k-hop neighborhood of a node n as the set of all nodes in the social network graph that are at most k hops away from n. n's 1-hop neighborhood consists of all of n's direct friends, and its 2-hop neighborhood consists of its direct friends and all of their first order friends.

We measure neighborhood sizes in our Facebook dataset for different hop lengths and plot the CDF in Figure 2. We see that more than half of all users have a 1-hop neighborhood size greater than 100. In addition, nearly 80% of the nodes have more than 100 FoFs in their 2-hop neighborhood, and nearly 70% of them have more than 1000 FoFs. With these large two-hop neighborhoods, users should be able to locate sufficient online friends to provide protection against attackers. We will examine that question in the next experiment.

One relevant question is, can we trust social links as real indicators of trust? While we did not perform user studies to quantify this, we did perform a high level experiment to understand the level of interaction between "friends" on Facebook. By examining user interactions on Wall posts and Photo albums, we conservatively estimate that an average Facebook user directly interacts with more than 1/3 of their friends list. More details are included in a forthcoming paper.

Availability of Friends. We next model the online availability of friends by understanding how many of the k-hop friends are available during the course of a typical anonymous session. Unfortunately, we are not aware of any studies on user session lengths in social networks. To drive our availability simulation, we use availability data obtained from two measurement traces, one of user activity on the Gnutella file-sharing network [15], and our own measurements of user online times and user session lengths on the Tor network.

[2]We obtained the dataset legitimately, and have been communicating with Facebook regarding our techniques and the dataset.

We gathered our Tor measurements by hosting and monitoring a Tor node for 1 week. We measured session lengths of more than 70,000 circuits going through our node, and plot a CDF in Figure 3. The average session length in our measurement is 853 seconds (14.2 minutes), which is consistent with a recent measurement study [11]. In addition, we also measured the availability of Tor nodes (node uptimes) by querying the Tor node directory. Along with the Gnutella data, we use these uptime numbers in our simulation of availability in the FoF neighborhood.

We ran our trace-driven availability experiment by mapping nodes in our experiment to nodes in the Gnutella and Tor datasets, using observed join and leave events in each dataset to drive node behavior in our network. We then select random users and measure the availability of their friends at random times (t) during our simulation. We group the availability results by the size of each user's FoF neighborhood size, and report the results in Table I. Each neighborhood size group has data from at least 500 data points. As we can clearly see, for users with FoF neighborhoods of size 1000 (more than 38% of Facebook users, as shown in Figure 2), more than half of them are likely to be online based on Tor data, and more than 25% of them are likely to be online based on Gnutella data. It has been observed that users in Gnutella exhibit frequent churn compared to other peer-to-peer applications [8], so we use the Gnutella results as a lower bound on our FoF availability results. Even so, we see that most users can expect hundreds of users in their FoF neighborhood to be online at any time.

FoF nodes can best shield a user from observation if they are online for the entire duration of a user session. We now quantify the percentage of online FoFs who will remain online for the entire duration of a 20-minute session. The 20-minute value is chosen as a conservative estimate based on our observations of Tor user sessions. For each of our random observations at time t, we compute the percentage of FoFs online at time t who remain continuously online at least until $t+20$ minutes. We plot the results in parenthesis in Table I. In both experiments driven by Gnutella and Tor, more than 80% of online FoFs remain online for the session duration, making them suitable as potential shields for the user.

Cliques in the Facebook Graph. We next quantify the number and size of user connectivity cliques in social networks. Searching for cliques is a known NP-hard problem, and the best tool we could find was Cliquer [14], which is still limited by its significant memory footprint. More specifically, even on a server with 32GB of RAM, Cliquer was limited to graphs of less than 1 million edges. Therefore, we ran Cliquer on a smaller dataset of 16,000 users from the Facebook Monterey Bay network.

Since we are interested in identifying large cliques to protect each user, we compute for each user the largest social clique that they are a member of. We plot the results as a CDF in Figure 4. Clearly, users vary significantly in their involvement in social cliques. The biggest observed cliques had 17 members. Roughly 58% of all users were members of cliques of size 3 or larger, and more than 18% of all users belonged to cliques of size 7 or larger. Note that by definition, each user connected to the social graph belongs to a clique of size 2 consisting of the user and their connected friend.

These results show that even against the modified two-phase logging attack, our path construction techniques can provide most users with reasonable anonymity sets that cannot be compromised by the attacker.

VI. Conclusions

In this paper, we propose and evaluate strategies to leverage "trusted" social links to protect anonymous communication from passive logging attacks. We investigate several approaches, propose a new two-phase logging attack on social anonymous networks, and analyze the robustness of our techniques against the attack. Our techniques prove resilient even when attackers learn the social network structure and compromise trusted friends. Finally, we use measurements of Tor, Facebook and Gnutella to show that our system can indeed improve anonymity significantly on today's anonymous networks.

Acknowledgments

We would like to thank Wil Robertson for providing access to the TOR measurement traces, and Bryce Boe and Christo Wilson for their work on gathering the Facebook measurement data.

References

[1] Ahn, Y.-Y., Han, S., Kwak, H., Moon, S., and Jeong, H. Analysis of topological characteristics of huge online social networking services. In *Proc. of WWW* (May 2007).

[2] The anonymizer. Anonymizer.com.

[3] Bauer, K., McCoy, D., Grunwald, D., Kohno, T., and Sicker, D. Low-resource routing attacks against tor. In *Proc. of WPES* (Alexandria, VA, 2007).

[4] Chaum, D. L. Untraceable electronic mail, return addresses, and digital pseudonyms. *CACM 24*, 2 (1981).

[5] Dingledine, R., Mathewson, N., and Syverson, P. Tor: The second-generation onion router. In *Proc. of USENIX Security* (2004).

[6] Freedman, M. J., and Morris, R. Tarzan: A peer-to-peer anonymizing network layer. In *Proc. of CCS* (Nov. 2002).

[7] Garriss, S., Kaminsky, M., Freedman, M. J., Karp, B., Mazieres, D., and Yu, H. Re: Reliable email. In *Proc. of NSDI* (San Jose, CA, May 2006).

[8] Guha, S., Daswani, N., and Jain, R. An experimental study of the skype peer-to-peer voip system. In *Proc. of IPTPS* (2006).

[9] Krishna P. N. Puttaswamy, A. S., and Zhao, B. Y. Improving anonymity using social links. Tech. Rep. 2008-13, UC Santa Barbara, Sept. 2008.

[10] Levine, B. N., Reiter, M. K., Wang, C., and Wright, M. Timing attacks in low-latency mix-based systems. In *Proceedings of Financial Cryptography* (Feb. 2004).

[11] McCoy, D., Bauer, K., Grunwald, D., Tabriz, P., and Sicker, D. Shining light in dark places: A study of anonymous network usage. Tech. Rep. CU-CS-1032-07, Univ. of CO, 2007.

[12] Mislove, A., Marcon, M., Gummadi, K. P., Druschel, P., and Bhattacharjee, B. Measurement and analysis of online social networks. In *Proc. of IMC* (Oct 2007).

[13] Mislove, A., Post, A., Gummadi, K. P., and Druschel, P. Ostra: Leverging trust to thwart unwanted communication. In *Proc. of NSDI* (April 2008).

[14] Ostergard, P. Cliquer - routines for clique searching. http://users.tkk.fi/~pat/cliquer.html.

[15] Saroiu, S., Gummadi, P. K., and Gribble, S. A measurement study of peer-to-peer file sharing systems. In *Proc. of MMCN* (January 2002).

[16] Swamynathan, G., Wilson, C., Boe, B., Almeroth, K. C., and Zhao, B. Y. Can social networks improve e-commerce: a study on social marketplaces. In *Proc. of WOSN* (August 2008).

[17] Syverson, P., Tsudik, G., Reed, M., and Landwehr, C. Towards an Analysis of Onion Routing Security. In *Proc. of WDIAU* (July 2000).

[18] Wilson, D. Limewire building social network. ZeroPaid News, January 2008.

[19] Wright, M., Adler, M., Levine, B. N., and Shields, C. An analysis of the degradation of anonymous protocols. In *Proc of NDSS* (Feb. 2002).

[20] Wright, M., Adler, M., Levine, B. N., and Shields, C. Defending anonymous communications against passive logging attacks. In *Proc. of IEEE Symposium on Security and Privacy* (May 2003).

[21] Wright, M., Adler, M., Levine, B. N., and Shields, C. The predecessor attack: An analysis of a threat to anonymous communications systems. *ACM TISS 7*, 4 (2004).

[22] Yu, H., Gibbons, P., Kaminsky, M., and Xiao, F. A near-optimal social network defense against sybil attacks. In *Proc. of IEEE Symposium on Security and Privacy* (Oakland, CA, May 2008).

[23] Zhuang, L., Zhou, F., Zhao, B. Y., and Rowstron, A. Cashmere: Resilient anonymous routing. In *Proc. of NSDI* (Boston, MA, May 2005).