Vulnerability in Socially-Informed Peer-to-Peer Systems

Jeremy Blackburn, Nicolas Kourtellis, Adriana Iamnitchi

Department of Computer Science and Engineering, University of South Florida, Tampa, FL, USA
{jhblackb, nkourtel}@mail.usf.edu,anda@cse.usf.edu

Abstract

The recent increase in the volume of recorded social interactions has the potential to enable a large class of innovative social applications and services. The decentralized management of such social information as a social graph distributed on a user-contributed peer-to-peer network is appealing due to privacy concerns. This paper studies the vulnerability of such a peer-to-peer system to attacks staged by malicious users who try to manipulate the graph or by malicious peers who try to manipulate the mining of the social graph. We discuss the effects and limitations of such attacks and we show experimentally how the distribution of the social data onto peers affects the system's resilience.

Categories and Subject Descriptors   H.3.4 [*Systems and Software*]: Distributed systems

General Terms   Experimentation, Security

Keywords   social graph, social data management, peer-to-peer, socially-aware applications, security

1.   Introduction

Socially-aware applications and services have leveraged out-of-band social relationships for diverse objectives such as improving security [Yu 2006], inferring trust [Maniatis 2005], providing incentives for resource sharing [Tran 2008], and building overlays [Popescu 2004] for private communication. Online social information has been used to rank Internet search results relative to the interests of a user's neighborhood in the social network [Gummadi 2006], to favor socially-connected users in a BitTorrent swarm [Pouwelse 2008], and to reduce unwanted communication [Mislove 2008]. Loopt, Brightkite, Foursquare, Google's Latitude, and others have combined social information and location/collocation into a new class of novel mobile applications.

The state of the art for such applications is to collect and manage social information in the form of a social graph, but within the context of the application. This approach inevitably reduces the accuracy of the social world representation and poses challenges such as application bootstrap. An alternative idea proposed in our previous work [Anderson 2010, Kourtellis 2010] is to aggregate social information from multiple sources in a multi-edged social graph, where edges are labeled corresponding to the type of social interaction they represent. Moreover, importing lessons from sociology, such edges should be directed and weighted, to reflect the fact that social relationships are asymmetrically reciprocal [Wellman 1988]. Weights may represent the level of interaction between users, as in [Chun 2008], or the strength of social relationships, such as how friends are related in their movie and music preferences [Lewis 2008].

Such a multi-edged, weighted and directed social graph comprises a more accurate representation of social relationships between users and can provide support to a variety of social applications. Applications can call the service that manages the social graph for social inferences such as finding the top contacts along a particular interaction (e.g., hiking), for objectives such as finding social incentives for resource sharing (e.g., storing hiking pictures), for content-aware communication (e.g., inviting hiking buddies and their hiking buddies to a trip), and others.

Due to the significant privacy concerns raised by aggregating such information from various sources as well as to the frequent privacy concerns brought by the business model that enables commercial online social networks such as Facebook, this social graph should be stored in a distributed manner on user-contributed resources. A user's social data, which consists of the user's relations with other users, could be stored (encrypted) on a peer-to-peer network. Peers are contributed by users, but unlike in traditional peer-to-peer networks, there is no need for each user to contribute a peer. Instead, *multiple* users can store their social data on a single peer, and an individual user's data can be replicated on a set of peers for reliability. Applications querying the social graph distributed on such a network will inevitably traverse the social graph and thus the peer-to-peer system that stores it.

Figure 1. An example of a multi-edged, labeled, directed, and weighted, social graph.

This paper examines the vulnerability of such a system to malicious attacks mounted by users trying to manipulate the social graph or by peers trying to manipulate the responses to requests that mine the social graph. Section 2 describes in detail the model considered in this work. Section 3 presents how malicious users and peers can attack the distributed social graph. We provide an experimental evaluation of the degree to which malicious peers can influence inferences on the social graph in Section 4 and we discuss the implications of these results in Section 5.

2. Model

Our model of a distributed, P2P social graph differs from traditional social graphs in two primary ways: 1) the structure of the social graph and 2) how it is distributed among peers. As mentioned before, the social graph is directed, weighted, labeled, and multi-edged. Each edge in the social graph represents a particular type of social connection between an *ego* and an *alter*. The label of the edge describes the type of social connection and the weight of the edge describes the intensity of the connection. The set of all edges originating from *ego* is *ego*'s *social data*.

For example, in Figure 1, *Ed* has an edge going to *Bob* describing the intensity of *Ed*'s "games" and "reads-blog" relationship with *Bob*. While *Bob* has a reciprocal "games" edge back to *Ed*, there is no reciprocal edge for "reads-blog". *Bob* also has an asymmetric "school" relationship with *Dave* as indicated by the different weights on their respective "school" edges. Social sensors, applications that aggregate and analyze the history of a user's interactions with other users, can be used to collect and report the social connections between users to populate the graph.

In our P2P model, a user's social data is stored on *at least* one peer, and each peer stores *at least* one user's social data. Each peer maintains the union of the social data of the users it represents. Depending on the social relationship of these users, the union can be anywhere from a disjoint set of edges to a connected subgraph. This differs from other models that have one or more dedicated peers per user [Buchegger 2009], or isolate each user's social data [Cutillo 2009, Shakimov 2008].

Figure 2. Peer-to-peer social bandwidth and distributed social graph topology. Large circles represent peers. Small circles represent users. Lines between peers represent social bandwidth. Lines between users represent the social ties connecting them.

Figure 2 visualizes the relationship between the P2P topology and the social graph. The large circles are peers. Small circles clustered around a peer represent users whose social data is stored on that peer. Thick lines connecting the peers represent the relations between social data stored on the peers, which we deem the "social bandwidth" between peers. The thin lines connecting users are the social ties between those users. For clarity, the social ties between users whose social data is stored on the same peer are not shown. Higher concentration of social ties between users who store social data on two peers directly translates to a higher social bandwidth between two peers in the P2P topology. Social bandwidth thus describes how a social graph traversal will translate to the P2P network.

This model of a distributed social graph enables a wide variety of applications. Social search, for example, is a method of connecting searchers to context appropriate content made available by their friends. An application performing a social search could follow social edges with contextually relevant labels over multiple social hops to users' content. A specific instance of social search is locating resources, e.g., disk space to store pictures from a hiking trip. Another set of applications can use this model for contextually-aware information dissemination. One can envision a P2P messaging system that routes messages based on social connections, in which a message is sent along contextually relevant and appropriately weighted social connections until reaching a specified number of destinations. A peer-to-peer money lending system, such as Prosper.com, could provide customized risk analysis based on social connections between potential lenders and borrowers, thus augmenting traditional risk analysis metrics.

What all such applications have in common is the routing of requests along selected social edges in the social graph. Since the social graph is distributed on top of a peer-to-peer network, these requests will be routed from peer to peer in a manner informed by the topology of the social graph. We call such peer-to-peer systems *socially-informed* because the communication pattern between peers is determined by the topology of the social graph and can be seen independently of the P2P overlay. Our previous work introduced Prometheus [Kourtellis 2010], an instance of such a socially-informed P2P system, that allows users to store their social data on selected peers. Users' social data is exposed through a set of social inferences, which are fulfilled via socially-informed routing between peers.

3. Vulnerability to Malicious Attacks

The ability of malicious attackers to affect the information that flows between peers in our model affects how useful it is to applications. Specifics of attacks, and the extent of the damage they can cause, differ with applications. An attack on a social search application might attempt to provide non-legitimate results by routing the request in a specific manner, perhaps distributing malware that appears to come from friends.

While the above is by no means exhaustive, in general, a socially-informed P2P topology faces two vectors for attacks: 1) manipulation of the social graph by malicious users and 2) manipulation of social inference requests by malicious peers. A user-based attack involves one (or more) malicious users attempting to manipulate the graph by creating, deleting, or modifying social edges. A peer-based attack involves one (or more) attacking peers manipulating social inference requests sent to other peers. In either case, an attack is successful if it affects queries involving legitimate users.

3.1 Malicious Users

Consider a malicious user *Alice* who attempts to create false relations with other users. Due to the directed graph model populated by social sensors that record Alice's interactions, she could, for example, email many users and thus enforce the creation of directed edges from her to them. While *Alice*'s 1-hop neighborhood could include the entire graph, she will not be a part of any other user's 1-hop (or indeed n-hop) neighborhood (unless those users contacted her). Without a reciprocal edge, *Alice* is not discoverable via the social data of other users, and thus cannot have any influence over them.

If *Alice* has a reciprocal edge between her and a legitimate user *Bob*, then she can affect queries that involve *Bob*'s relations as she is in his 1-hop neighborhood. Since *Bob*'s 2-hop neighborhood now includes *Alice*'s 1-hop neighborhood, she could introduce other malicious users to legitimate users via *Bob*'s neighborhood.

Our labeled, weighted, multi-edged social graph model provides two mitigating factors to this type of attack, allowing inferences to limit *Alice*'s success to requests involving 1) the label of the reciprocal edge she has with *Bob* and 2) a minimum edge weight. For an attack to be successful, *Alice* must not only manage to get a reciprocal edge created (which means an interaction initiated or recognized by Bob), but this reciprocal edge must be of the proper label and have an appropriate minimum weight. Because the reciprocal edge is not under *Alice*'s direct control, she needs to maintain what amounts to a legitimate relationship with a legitimate user over a period of time to have any meaningful effect on queries involving that user.

Finally, there is a more subtle attack that a user can stage by manipulating an outgoing social edge. Consider requests that traverse the social graph over different types of edges and weights to reach users that are directly or indirectly connected, over multiple social paths [Kourtellis 2010]. A malicious user could manipulate such complex requests that search for the social path between two users with the highest overall weight. If *Alice* has a reciprocal edge with *Bob* and creates a fake edge with high weight to *Josh*, as in Figure 1, she can mislead such a request originating from *Bob* to utilize the indirect social path of *Bob* to *Josh* through her, instead of the direct path between the two users.

3.2 Malicious Peers

A malicious peer has several mechanisms for attacking requests. For example, modifying results sent to other peers, dropping requests, changing the parameters or type of a request, or creating fake requests. These attacks are difficult to detect. We note that due to the distributed nature of the social graph, there is no way to verify the validity of the results returned by any given peer.

If a peer does not have the social data necessary to fulfill a request locally, then a different peer must provide it. Furthermore, in a system that protects user privacy, a requestor cannot distinguish how and from what peer any given item entered the result set. Thus, if a malicious peer serves the ith hop of a n-hop request, it can "override" the results of the ith $+1$ to the nth hop of its leg of the request and remain undetected. In the following section we evaluate the effects of such a peer based attack.

4. Experimental Evaluation

We study system vulnerability to malicious peers by measuring peers' opportunity to influence results when serving a *neighborhood* inference request. A neighborhood inference traverses the social graph in a breadth-first manner starting from a source user, following only edges with a particular type of label and a minimum weight, and within a radius n social hops from the source. In these experiments we consider the worst case scenario in which we do not restrict the weight of the social relationships to be considered and

all edges are reciprocal. A peer's *influence* is the fraction of requests that the peer serviced over the total number of requests issued during the experiment. This fraction represents the overall opportunity of a peer to manipulate the results of any given request issued.

4.1 Experimental Setup

We used a synthetic social graph of 1000 users created by a synthetic social network generator based on the model introduced in [Vázquez 2003] and refined in [Sala 2010]. We considered these users mapped onto 100 peers. In the context of these experiments, we consider a user "mapped" on a peer when the user's social data is stored on that peer. A user's data can be replicated on K peers on average. Finally, a peer can store the social data of N users on average, i.e., N users are mapped on a peer.

Intuitively, the influence of a peer increases with the number of social hops the request is issued for. In particular, since an n-hop request is served by up to n peers, as n approaches the diameter of the graph, the influence of any given peer approaches 100%. The choice of a 1000 user social graph allows us to reach nearly 100% of the users in the graph within 4 hops, exceeding the "horizon of observability" [Friedkin 1983] by only 2 hops. We do not consider the first hop (i.e., the source peer) of a request as malicious, since if it is, no results can be considered legitimate. Thus, we measured influence for $n = 2, 3$ and 4 hop requests.

During the experiments, an n-hop neighborhood request is performed for each *ego* (user). The peer that serves the request is randomly selected from the peers the *ego* is mapped to. We call this peer P_0. The set of users that appear in the ith hop of an n-hop request is r_i. For example, *ego*'s r_1 contains the users that appear in *ego*'s social data, i.e., the 1-hop neighborhood of *ego*. For every user in *ego*'s r_1 *not* mapped to P_0, a peer P_m is randomly selected from the set of peers storing that user's social data to serve the next hop of the request, with P_0 serving the next hop for users that *are* mapped to it. Each $P_{m \neq 0}$'s influence increases when it serves a request. For example, in Figure 2, let us assume that a 2-hop neighborhood request for user 8 is sent to peer 66. This peer will check the 1-hop social connections of user 8 and will forward secondary neighborhood requests to the peers these connections are mapped to. So if users 1 and 7 are within these connections, a secondary request will be sent to each of peers 84 and 16 respectively. Each of the peers receiving such requests increase their total influence in the experimental run.

To eliminate any bias introduced by the random peer selection described above, we performed $T = 100$ iterations of each experimental configuration and verified that the average numbers reported in the results are within tight 95% confidence intervals. A summary of the various parameters and values used during the experiments is shown in Table 1.

The distribution of each user's social data on peers directly affects the topology of the requests traversing the P2P

Table 1. Values of parameters used during experiments.

Parameter	Value
N (users per peer)	10, 20, 40, 50
K (peers per user)	1, 2, 4, 5
n (social hops)	2, 3, 4
T (iterations)	100
Mapping (user-peer)	Random, Social

network. We consider two different user-peer mappings, one random and one social, as explained next. In both cases, on average, N users are mapped on a peer and each user's data is replicated onto K peers.

In the random mapping, users' social data is stored on randomly selected peers. Consequently, groups of random users are mapped on the same peer.
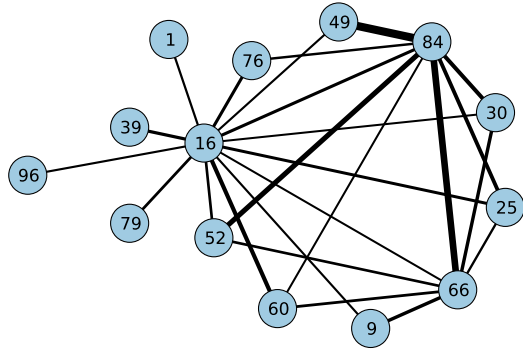
The social mapping corresponds to a more realistic scenario, in which a group of socially-connected users share the resources provided by a peer (potentially contributed by a member of the group). In our experiments, we created such a social mapping by using a modified version of the community detection algorithm introduced in [Girvan 2002]. The algorithm takes as input a social graph, the number of communities to be identified (which in our case is the number of peers in the system) and the minimum acceptable community size. The algorithm recursively removes the social edge with the highest edge betweenness centrality if by removing it a new social community of the desired size is created. Removal of edges continues until the specified number of communities is met. Users from a given community are then mapped onto the same peer.

Instantiations of the P2P topology formed by a social and a random mapping of the 1000 users (from our synthetic graph) onto 100 peers, with $N = 10$ users per peer, and $K = 1$ peer per user, are shown in Figures 3(a) and 3(b), respectively. To make the topologies more visible, we present each mapping's P2P topology formed by the 25 highest social bandwidth connections. We observe that the P2P topology gains structure under a socially-aware mapping in comparison to the random mapping. Note also that the connections between peers in the social mapping have differing social bandwidths versus the relatively uniform social bandwidth connections of the random mapping.
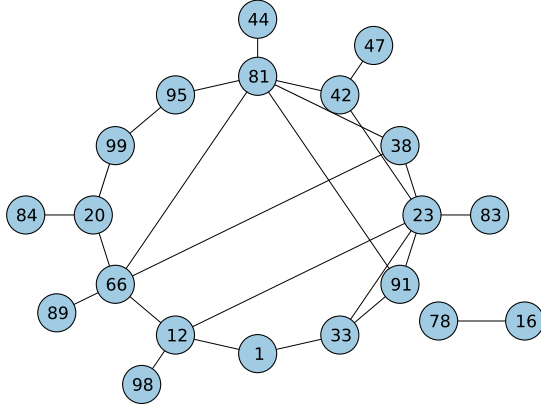
4.2 Results

The average peer influence is plotted in Figure 4 with the maximum and minimum influence as error bars. From these results we make several observations.

The social mapping results in a more resilient P2P system. This is because in a social mapping, socially close users are mapped to the same peers, increasing the likelihood that more hops of an n-hop request will be served locally when compared to randomly mapped users. The average influence of peers in the random mapping was about 5% more than the
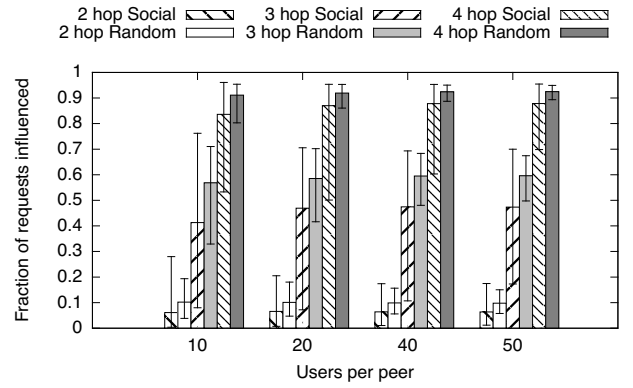
(a) Social mapping using communities

(b) Random mapping

Figure 3. The P2P topology formed by the 25 highest social bandwidth connections between peers. The thickness of the lines between peers is directly proportional to the social bandwidth between them.

Figure 4. Average peer influence. The maximum and minimum influence of peers are plotted as error bars.

mapped onto peers. This is due to users of high social degree being closely connected and more likely to be mapped together on the same peer. For example, for $N = 20$ in the social mapping, the average sum of social degrees of users mapped to a peer is about 300. There are, however, several peers whose mapped users have a cumulative social degree of over 600 with 2 particular peers having a cumulative social degree of over 1100. These peers account for the disparity in maximum and average peer influence. It is important to note that while the social mapping does have maximums well above the average, the maximum values are still within a few percent of the maximum values for the random mapping.

5. Summary and Discussion

The increasing number of socially-aware applications and services, the privacy concerns that go along with the requisite exposure of social information they use, and lessons on relational representation from sociology lend credence to the need for a *directed*, *weighted*, *labeled*, and *multi-edged* social graph distributed on a P2P storage system. In this paper we qualitatively analyzed the resilience of such a socially-informed P2P topology to malicious users manipulating the underlying social graph. While edge creation is cheap, we have demonstrated that the underlying representation of the social graph provides barriers to malicious users. We also discussed the subtle effects user manipulation might have on complex graph traversals, such as those introduced by the inference requests in [Kourtellis 2010].

From this analysis we learned that inference requests, and other applications, must be carefully designed to reduce the impact of malicious users. The structural attributes of the social graph should be used to mitigate the effects of malicious users. In particular, edge direction should be taken into account to force malicious users to have a reciprocal edge with a legitimate user. Edge weight and label should be used to differentiate edges, in turn forcing malicious users to

social mapping for 2-hop requests. This further increased to around 15% for 3-hop requests. For 4-hop requests, the gap is reduced back to 5%. This reduction in influence differentiation is due to the properties of the graph: the average path length is about 3.5, which results in nearly all users being in a 4-hop result set, in turn increasing the likelihood of all peers having some influence on a request. Even so, the social mapping still managed to outperform the random mapping. We also note that while there is an increase in average influence in the social mapping from $N = 10$ to $N = 20$, further increases in replication do not significantly affect vulnerability.

While a socially mapped system is less vulnerable on average, the most influential peer in the social mapping is able to influence about 8% more 2-hop requests than the most influential peer in the random mapping when $N = 10$. This difference decreases when $N \geq 20$. The maximum influence of each mapping is significantly higher than the average for all N and n-hop requests; especially for the social mapping. This observation indicates that there are "high value" peers to target for malicious attacks when users are socially

not only have reciprocal edges, but *contextually meaningful* reciprocal edges.

Our experimental evaluation demonstrates that a distributed, P2P social graph's vulnerability to attacks by malicious peers is affected by how the traversal of the P2P network is informed by the underlying social graph. We summarize these findings into three lessons:

1) A social mapping of users to peers reduces the average influence of peers. In such mappings, the social data stored on a peer act as a subgraph of the global social graph. As the social graph is traversed, this leads to a reduction in the number of peers involved, because the local social data enable a single peer to fulfill multiple consecutive hops of the traversal. This effect is visible for 2-hop requests, intensifies for 3-hop requests, and is the reason why the average influence of peers for the social mapping is lower than for the random mapping.

In a larger graph, we expect the *absolute* influence of peers to decrease because more hops are necessary to reach the majority of the users. However, the *relative* difference in influence should follow the results from our smaller graph.

2) The average influence of peers is not significantly affected by the replication of users' social data.

3) The existence of influential users in the social graph translates into higher network vulnerability if the peers that represent them are targeted for malicious attacks. This is a consequence of any mapping that attempts to place well connected users together. Since high degree users tend to be associated, they are more likely to be mapped together resulting in the emergence of *hub peers*. Their social data increase the social bandwidth of the peer they are mapped onto and results in that peer having greater influence over traversal of the social graph.

Acknowledgments

References

[Anderson 2010] P. Anderson, N. Kourtellis, J. Finnis, and A. Iamnitchi. On Managing Social Data for Enabling Socially-Aware Applications and Services. *3rd ACM EuroSys Workshop on Social Network Systems*, 2010.

[Buchegger 2009] S Buchegger, D Schiöberg, LH Vu, and A Datta. Peerson: P2p social networking: early experiences and insights. *2nd ACM EuroSys Workshop on Social Network Systems*, 2009.

[Chun 2008] H. Chun, H. Kwak, Y-H Eom, Y-Y Ahn, S. Moon, and H. Jeong. Comparison of online social relations in volume vs. interaction: a case study of Cyworld. *8th Conference on Internet Measurement*, pages 57–70, 2008.

[Cutillo 2009] LA Cutillo, R Molva, and T Strufe. Safebook: a privacy preserving online social network leveraging on real-life trust. *IEEE Comm. Magazine*, 2009.

[Friedkin 1983] N. E. Friedkin. Horizons of observability and limits of informal control in organizations. *Social Forces*, 62 (1):57–77, 1983.

[Girvan 2002] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *National Academy of Sciences of USA*, 99(12):7821–7826, June 2002.

[Gummadi 2006] K. P. Gummadi, A. Mislove, and P. Druschel. Exploiting social networks for internet search. *5th Workshop on Hot Topics in Networks*, pages 79–84, 2006.

[Kourtellis 2010] N. Kourtellis, J. Finnis, P. Anderson, J. Blackburn, C. Borcea, and A. Iamnitchi. Prometheus: User-controlled p2p social data management for socially-aware applications. *11th International Middleware Conference*, November 2010.

[Lewis 2008] K. Lewis, J. Kaufman, M. Gonzalez, A. Wimmer, and N. Christakis. Tastes, ties, and time: A new social network dataset using Facebook.com. *Social Networks*, 30(4):330 – 342, 2008.

[Maniatis 2005] P. Maniatis, M. Roussopoulos, T. J. Giuli, D. S. H. Rosenthal, and M. Baker. The LOCKSS peer-to-peer digital preservation system. *ACM Trans. Comput. Syst.*, 23(1), 2005.

[Mislove 2008] A. Mislove, A. Post, P. Druschel, and K. P. Gummadi. Ostra: leveraging trust to thwart unwanted communication. *5th Symposium on Networked Systems Design and Implementation*, pages 15–30, 2008.

[Popescu 2004] B. C. Popescu, B. Crispo, and A. S. Tanenbaum. Safe and private data sharing with turtle: Friends team-up and beat the system. *12th International Workshop on Security Protocols*, pages 213–220, 2004.

[Pouwelse 2008] J. Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D. H. J. Epema, M. Reinders, M. van Steen, and H. Sips. Tribler: A social-based peer-to-peer system. *Concurrency and Computation: Practice and Experience*, 20:127–138, 2008.

[Sala 2010] A. Sala, L. Cao, C. Wilson, R. Zablit, H. Zheng, and B. Y. Zhao. Measurement-calibrated graph models for social network experiments. *19th International World Wide Web Conference*, April 2010.

[Shakimov 2008] A Shakimov, H Lim, LP Cox, and R Cáceres. Vis-à-vis: Online social networking via virtual individual servers. *Duke University Technical Report TR-2008-05*, 2008.

[Tran 2008] D. N. Tran, F. Chiang, and J. Li. Friendstore: cooperative online backup using trusted nodes. *1st Workshop on Social Network Systems*, pages 37–42, 2008.

[Vázquez 2003] A. Vázquez. Growing network with local rules: Preferential attachment, clustering hierarchy, and degree correlations. *Physical Review E*, 67(5), May 2003.

[Wellman 1988] B. Wellman. Structural analysis: From method and metaphor to theory and substance. *Social structures: A network approach.*, pages 19–61, 1988.

[Yu 2006] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. Sybilguard: defending against sybil attacks via social networks. *SIGCOMM'06 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 267–278, 2006.