Pass it on: Social Networks Stymie Censors

Yair Sovran, Alana Libonati, Jinyang Li
New York University

Abstract

Many countries exploit control over the Internet infrastructure to block access to “grey” materials. One common way to access blocked contents is to relay traffic via an unblocked proxy operating outside the censored domain. This paper discusses the challenges facing any proxy-based circumvention system and argues that a successful system should disseminate proxies’ addresses to legitimate users while shielding the addresses from the censor who, posing as a user, could learn of and block the proxies themselves. We propose Kaleidoscope, a circumvention system that disseminates proxy addresses over a social network whose links correspond to existing real world social relationships among users. Kaleidoscope ensures each node learns only a small, consistent subset of the proxies. Because the censor is unlikely to subvert a large fraction of the social graph, he is not able to learn of, and thus block, a large number of proxies.

1 Introduction

The Internet was originally designed to ensure robust communication in the face of attacks on the communication infrastructure itself. A popular quote says “the Internet treats censorship as a malfunction and routes around it”. On the contrary, Internet censorship is prevalent today. Empirical studies reveal that millions of users in many parts of the world suffer from Internet censorship [4] and news have reported the blockage of a variety of websites ranging from YouTube [27], Flickr [6], Wikipedia [6] or even Google [2].

Censorship is not simply a technical problem. Although censors often employ technical methods, access to many types of censored material is prevented via non-technical methods such as the threat of detention, job loss, etc. Understanding the full range of methods (both technical and non-technical) available to the censor serves to outline the potential scope of any technical solution to the problem of censorship and informs the design of such a system.

Figure 1 gives a rough classification of approaches to censorship. Censored materials are plotted according to the techniques used to prevent access to the materials. The x-axis corresponds to the degree of technical means employed (IP address blocking, packet inspection, etc.); the y-axis corresponds to non-technical approaches (threat of jail terms, violence). The materials can be roughly grouped into four quadrants.

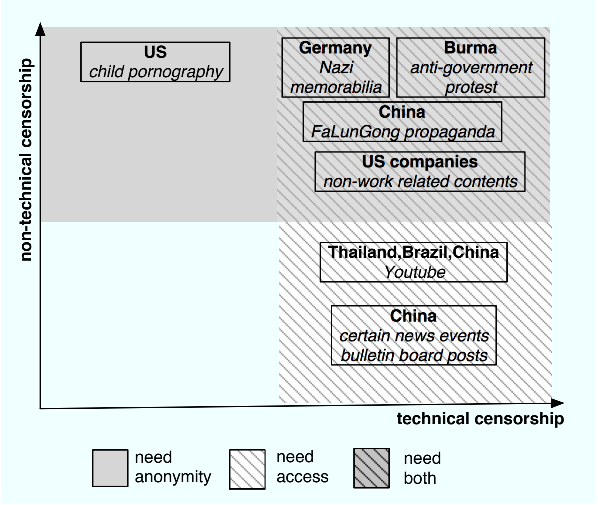


Figure 1: A classification of approaches to censorship with example censored materials.

At the top right, censorship is enforced via both technical means as well as various degrees of real world punishments. For example, during recent political protests, the Burmese government went to extremes of both technical and non-technical censorship. The government temporarily disrupted the entire country’s external Internet connectivity and also detained those people who used outside proxies to post censored materials on otherwise blocked websites [5]. At the top left lie situations where censorship is only enforced via non-technical methods. For example, the US government does not block access to overseas child pornography sites but does prosecute individuals for viewing them under statutes that make possessing such material illegal.

Interestingly, there is a large amount of content blocked via purely technical means for which there is no associated real world punishments for accessing (although such punishments still exist for publishers). These are contents that lie to the bottom right quadrant in Figure 1. Even though it seems odd that such "grey" materials exist in the first place, they are common in countries where the government maintains tight control of the news media. For example, a recently disclosed internal article has shown that the Chinese government blocks many news articles before they are approved as official and it also requires websites to block bulletin board posts that have generated too many comments [8]. Furthermore, because technical censorship is never precise, many blocked contents are simply the result of collateral damage as in the case of the blocking of Google [2], YouTube [27], and Flickr [6]. Another recent study of the Chinese firewall has also found that many banned keywords are quite innocuous such as "student union", "international geological society" [14].

Figure 1 also demonstrates that there are two distinct problems to be addressed in defeating censors: access and anonymity. Given the very real penalties at stake for users, any system designed to operate in the upper two quadrants of Figure 1 (those protected by non-technical means) must guarantee anonymity for readers. Any system designed to operate in the right two quadrants must provide access to material despite technical hurdles (materials in the top-right quadrant require both access despite blocks and anonymity for users).

We will focus on the problem of grey materials in the lower-right quadrant of Figure 1. Building systems to provide access for grey materials avoids many of the complications of systems designed to provide access to censored materials in the top two quadrants: since there is no real world punishment for viewing or helping others view these grey materials, a circumvention system can safely run on users' desktops and even enlist volunteer helpers both outside and inside the censored domain to relay traffic for others. While the system we consider won't address either of the top two quadrants, anonymity-guaranteeing systems have been built (Tor, for example) and provide a possible solution for material in the top-left quadrant. Systems like Tor do not provide a solution for the upper-right quadrant of our figure; combining an anonymity system like Tor with a blocking circumvention system might be a solution for this part of the problem space. Such a combination is left to future work.

2 Challenges

There are three primary methods currently used by censors to block direct access to grey materials [4]: DNS poisoning, IP address blocking and selective resetting of TCP connections [13]. These techniques exploit the fact that the censor typically controls the underlying communication infrastructure, giving him the ability to inspect packets inside the censored domain; these packets can be dropped if they are addressed to a banned destination (IP address blocking) or inspected for banned keywords (TCP resetting, DNS poisoning).

Although direct communication is blocked, users can access banned sites via unblocked proxies located outside of the censor's reach (in a different country for example). This is the most common approach to circumventing blocks on grey material and the one we'll focus on in this paper. To use a proxy, some care must be taken: to avoid content-based blocking, the connection to the proxy must be encrypted; using a DNS server outside of the censored domain avoids the problem of DNS poisoning. The final, and most difficult, challenge is preventing the censor from simply adding the proxies' addresses to the list of blocked sites. Additionally, the censor could also pose as a legitimate proxy to learn about clients wanting access to banned sites and potentially block these clients as well. This is the primary challenge addressed by this work: how can we make legitimate proxies and users aware of each other, but not to the censor, even when the censor is able to pose as a legitimate user or a proxy?

Existing systems have already proven vulnerable to this attack. Many proxy services use a centralized website to help users learn about proxy addresses; the censor can simply block such centralized discovery services. The Anonymizer [1] and SafeWeb [9] systems were disabled in this way in China. Furthermore, if a system reveals too many proxy addresses indiscriminately to any user that requests them, the censor can join the system to learn about all proxies and subsequently block them. For example, Wikipedia has discovered and blocked all 700+ Tor anonymizing relay servers to prevent Tor users from anonymously defacing Wikipedia pages [15]. The censor could also pose as a legitimate proxy to serve unsuspecting users in order to track or possibly block them. A recent study of the Gnutella network has shown that almost all users are tracked by a few nodes that are suspected of colluding with the RIAA [10]. As these incidents show, the censor can effectively disrupt service by blocking any centralized components used for software distribution, resource discovery, etc. Even if the system is fully decentralized, the censor could block all proxies or those users seeking proxy services if he can reliably find out about their addresses.

These attacks provide guidelines for the design of any system to resist blocking of gray material using proxies. The system must be fully decentralized; if the system depends on any single, publicly-known host, that host will be blocked by the censor and the system will be disabled. A distributed system must enlist the service of many proxies and also partition knowledge of the proxies among the participating nodes: if any node can learn the identity of every proxy, the censor can as well and will block all of them. Furthermore, each proxy should be allowed to only directly serve a small number of users: if any proxy is allowed to directly serve all users, the censor can pose as a legitimate proxy to learn about all users and possibly block them. Finally, the system must not expect to be able to provide service to all users when under attack: the censor will inevitably join the system and block some proxies, disrupting the system for some subset of legitimate users. These disrupted users must, unfortunately, not be allowed to learn the identity of new proxies: if they were allowed to, the censor could as well. Repeating this process would allow the censor to block all proxies in the system.

Finally, since the adversary controls the underlying communication infrastructure, we realize that attempting to elude the most determined censor's attempts to block our access to material is fruitless: a censor could always deploy the "nuclear option" of pulling the plug on all external traffic as the junta ruling Burma did. A slightly less drastic approach might be to drop all encrypted traffic making it impossible to avoid content filters or white-listing the set of allowed remote sites instead of black-listing banned sites. A determined censor could also deploy sophisticated traffic analysis to thwart blocking: by observing all known participating nodes simultaneously, he may be able to learn the identity of additional, unknown nodes by identifying nodes with similar traffic patterns. Although we can't hope to defeat an adversary as powerful and ruthless as the ones we imagine here, a system which forces the censor to implement increasingly costly measures is still useful. If the censor is forced to implement strategies that affect legitimate traffic (blocking encrypted traffic also affects online commerce, for instance) or to undertake increasingly complex analysis schemes, the censor is likely to judge that the gain to be had in blocking these grey materials does not justify the required cost (a cost that our system increases).

3 Kaleidoscope Design

The previous section outlined the challenges and basic requirements for a proxy-based circumvention system. This section shows how our proposed system Kaleidoscope addresses those challenges.

Kaleidoscope allows users inside the censored domain to access blocked websites by relaying traffic via volunteer proxies residing outside the censored domain. To prevent the censor from learning the addresses of all users or proxies, Kaleidoscope distributes a small random subset of proxy identities to participants. If the censor could only learn a small subset of proxies by joining the system, most legitimate users still have some proxy not known to the censor. Kaleidoscope does not guarantee access for all users: some users will find that all of their proxies have been blocked and will not be able to gain access.

This design requires that the censor can not join the network an arbitrarily large number of times and collect a large number of random subsets to eventually discover all proxies. To prevent this attack we take advantage of the trust relationships among users: nodes only distribute information about proxies to nodes that are owned by people that the node owner has reason to trust (because of pre-existing real world social relationships). By equating the social network of users with the proxy distribution network, Kaleidoscope requires the censor to subvert real-world trust relationships to obtain additional vantage points on the system.

If the social network is to effectively limit the censor's ability to join the network, users must establish trust links carefully. The trust links we envision are not the same as those displayed on today's popular social networking sites. There is no disincentive to endorsing casual acquaintances or even strangers on these sites. In contrast, there are strong disincentives to choosing links carelessly in Kaleidoscope: subverted trust links cause a node to lose its known proxies and expose itself to the prying eye of the censor. The system could discourage casual links by artificially limiting the number of links a user may establish. We also allow users to refuse requests to create a link without alerting the requester so that peer pressure does not lead to ill-chosen links.

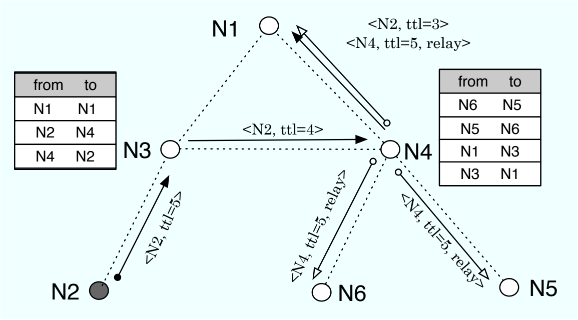
The most straightforward way to utilize the trust graph is to allow nodes to learn only the addresses of its immediate neighbors. This strategy incurs the least risk:

with one subverted trust link, the adversary can discover at most one proxy. Unfortunately, the scheme limits each proxy to serving only a few trusted neighbors and leaves many users without access to any proxy: most users do not have any proxies in their immediate neighborhood as users' neighbors are biased towards those in the same geographical region [22].

There are two basic techniques to increase the likelihood of nodes reaching a proxy. The first technique is to allow nodes to route requests to proxies over multiple hops in the underlying trust graph by asking nodes to relay requests. This allows a node without a proxy in its immediate neighborhood to obtain service. However, because peer nodes are not highly available, a node is unlikely to have any working multi-hop route to a proxy. Alternatively we could route information about proxy addresses over the trust network (instead of requests); nodes can then directly contact the proxies they learn of. This approach incurs additional risk since the censor could also join the network as a proxy and learn the addresses of clients: if each proxy is allowed to advertise itself to 2000 users, the adversary can also discover 2000 users and block their outgoing access. Thus, even if the adversary only manages to subvert a small fraction of trust links, he could significantly increase his knowledge of the network by advertising its service to a large number of users.

Kaleidoscope combines both of these techniques in moderation: each node advertises its address only to a small number (r) of other nodes beyond its immediate neighborhood (using a mechanism we describe below). To allow each proxy to service more users without directly learning their addresses, Kaleidoscope also forwards traffic to proxies via at most one intermediate relay node.

One approach to advertising a node's address to r other nodes would be to use a random walk of length r over the trust graph. This approach has several problems: first, if the walk is truly random, as the advertisement is repeated a new set of r nodes will learn the address each time. We can fix this problem by making the walk random, but repeatable: each time a node forwards an advertisement it remembers the next-hop node and uses the same next-hop node for all future advertisements from the same originator. This scheme also has a short-coming: if the censor is able to obtain a single link in the trust graph, he can produce an arbitrary number of identities "behind" that link. Each of these identities can advertise itself to r nodes as a proxy, allowing the censor to learn the identities of a large number of client nodes that is exponential with r. To prevent this attack, we use random routes [32] instead of random walks. A random route forwards a message to a pre-determined outgoing link based on the incoming link of the message (instead of the message originator's identity as in repeatable random walks). Regardless of how many identities the censor generates, he still controls only one link into the graph and thus can only reach at most r other nodes. The original random route is not symmetric [32]; we enforce symmetric routes so that the set of nodes that learn of a node and that a node learns of is identical (and has size r). This is important since a node can do damage both by being learned of (posing as a proxy) and learning of (blocking proxies).



from	to
N1	N1
N2	N4
N4	N2

from	to
N6	N5
N5	N6
N1	N3
N3	N1

Figure 2: Each node advertises its address-cookie pair along a random route of 5 hops via each of its neighbors. A relay node includes a `relay` flag identifying it as a relay. $N2$ is the only proxy. Arrows with a circle at one end denotes an original (instead of forwarded) advertisement.

3.1 Protocol Operation

Figure 2 gives an example of how relays and proxies advertise their addresses in Kaleidoscope. Node $N2$ is a proxy while all other nodes are relays. To construct symmetric random routes, each node randomly pairs its neighbors with each other so that it forwards a message received from one neighbor to the other one in the pair. Figure 2 shows node $N3$ and $N4$'s routing tables. Because $N3$ has an odd number of neighbors, it has to pair $N1$ with itself, causing $N3$ to route all all messages received from the singleton neighbor back to itself.

The underlying trust graph might change slowly over time as new users join the system and establish trust links. Incorporating changes in the trust graph into an existing routing table is straightforward: if $N3$ establishes a trust link with a new node $N7$, it can pair up any singleton neighbor (e.g. $N1$) with $N7$. Similarly, if $N4$ deletes the trust link to $N5$, it pairs $N6$ with itself.

When a node has more than one singleton neighbors, it pairs them together to create new routing entries.

Each proxy originates an advertisement via each of its neighbors. The advertisement includes its current address and a cookie to be used for access. The originator limits the random route length by setting the time-to-live field to a small number (ttl=r). We choose r=5 as a system wide parameter. If the censor advertises messages with the time-to-live field set to be more than 5, these messages will be dropped. Each relay node also originates advertisements including its own address-cookie pair, and a flag marking it as a relay node. This flag allows clients to choose a direct connection to a proxy over a relayed one when both are available. In Figure 2, proxy $N2$'s advertisement follows the path $N2 \rightarrow N3 \rightarrow N4 \rightarrow N1$... The relay node $N3$ advertises its address-cookie pair along three random routes via $N1, N5, N6$. As a result of these advertisements, $N3, N4, N1$ are able to learn of $N2$'s address for direct communication while $N6, N5$ can communicate with proxy $N2$ by forwarding encrypted traffic via $N4$.

3.2 Analysis

This section considers how the values for several of the system parameters affects Kaleidoscope: how many relay hops to allow, how many nodes a proxy should advertise itself to, and how many proxies are necessary to ensure service.

The length of the random route (r) that nodes use to advertise their identity reflects a tradeoff between reaching additional proxies and enhancing the censor's ability to collect information. If r is too small, the generated random route is unlikely to escape the censored domain (given the geographic clumping properties of social graphs, this is especially true). On the other hand, a longer random route allows the censor to discover correspondingly more node identities. If the censor could join Kaleidoscope via f subverted trust links, he is able to learn of rf users or proxies: Kaleidoscope increases the number of users he is able to discover by r. Simulations reveal that $r = 5$ represents a reasonable tradeoff.

We have simulated Kaleidoscope's performance using a synthetic trust graph of one million nodes according to the social graph model proposed in [28]. The average node degree in the synthetic graph is 4.65 and the maximum is 13. When 1.5% of nodes act as proxies, the median number of proxies each node can access indirectly is 3 and more than 90% of nodes know how to access at least one proxy. We have also experimented with a crawled Orkut social graph among 42474 users. To enable comparison with the synthetic graph, we only use a subset of the Orkut links by discarding links to enforce a maximum node degree of 13. Approximately 90% of crawled users reside in Brazil and we designate these nodes as being inside the censored domain. When 15% of nodes outside the censored domain act as proxies (the fraction of proxies is 1.5% of all nodes), the median number of proxies each node can access (directly or indirectly) is 7. This number is higher than that observed in the synthetic graph because the Orkut subgraph has a higher average node degree (5.59 vs. 4.65) and the neighbors of an Orkut node are less likely to be neighbors themselves. Hence, each random route in Orkut is likely to visit more distinct nodes than that in the synthetic graph, causing nodes to learn more proxies and relays. Even if the number of subverted trust links reaches half as many as the total number of proxies, more than 94% users can still access a proxy not known to the censor.

4 Related Work

Existing anonymity systems (e.g. [17, 25]) and censorship-resistant publishing systems (e.g. [12, 29] do not deal with an adversary that tries to block access to the system. Recently, the developers of Tor have proposed ways to make Tor resilient to such blocking attacks [16]. The proposed solution employs a centralized discovery service to disseminate a restricted set of relay identities to requesting users using multiple strategies. Another centralized discovery mechanism, key-space hopping [21] uses the same insights of restricting each user to discovering a limited set of proxies. Kaleidoscope's decentralized approach compliments both services since both need a user to relay his requests via some bootstrap proxy to contact the blocked central discovery server to learn about more proxies.

Many proxies are built to relay traffic to blocked websites, e.g. Infranet [20], Psiphon [7], Circumventor [3]. Many proxy-based circumvention systems rely on the owners of volunteer proxies to manually disseminate a proxy's identity to her friends. Kaleidoscope's use of the social network mirrors this manual dissemination but allows users to discover volunteer proxies that are further away in the trust graph. The use of social links for trust also resembles SPKI/SDSI [19] and PGP certification chains [33].

Kaleidoscope uses the trust network to guard against a censor who creates an arbitrary number of identities; this is, in essence, a Sybil attack [18]. Reputation systems [11, 23, 24, 26, 30, 31] might be used to detect censors who have infiltrated the system; however, since a proxy reveals its address to many nodes, it is hard to put the blame on any particular node when it is blocked.

5 Conclusion

Kaleidoscope is a peer-to-peer system that uses trust relationships among users to disseminate proxies and relays' addresses in order to help users discover and forward traffic to proxies to circumvent communication blockage. By controlling the number of nodes that can learn of each node's address, Kaleidoscope prevents a censor who only controls a limited number of trust links to learn of most nodes' addresses and block them. We believe Kaleidoscope is a promising solution to circumvent content blocking on the Internet today.

Acknowledgments

We thank Frank Dabek for helping us improve the design and the paper. Nikolaos Michalakis, Eric Hielscher and Robert Grimm have also contributed to the design. We are grateful to Robert Morris, Frans Kaashoek, David Bindel and the anonymous reviewers for their insightful comments.

References

[1] Anonymizer. `http://anonymizer.com/`.

[2] Bbc news. china criticised for ban on google. `http://news.bbc.co.uk/1/hi/technology/2238236.stm`.

[3] Citizens Lab. Everyone's guide to bypassing internet censorship. `http://deibert.citizenlab.org/Circ_guide.pdf`.

[4] OpenNet Initiative. `http://www.opennet.net`.

[5] Opennet initiative. pulling the plug: A technical review of the internet shutdown in burma. `http://opennet.net/research/bulletins/013/`.

[6] PC world. China blocks youtube, restores flickr and blogspot. `http://www.pcworld.com/article/id,138599-c,sites/article.html`.

[7] Psiphon. `http://psiphone.civisec.org`.

[8] Reporters sans border. Journey to the heart of internet censorship. `http://www.rsf.org/article.php3?id_article=23924`.

[9] Safeweb privacy proxy censored in china. `http://censorware.net/articles/01/03/14/0755209.shtml`.

[10] A. Banerjee, M. Faloutsos, and L. Bhuyan. Is someone tracking p2p users? In *IFIP Networking*, 2007.

[11] S. Buchegger and J.-Y. L. Boudec. A robust reputation system for p2p and mobile ad-hoc networks. In *Workshop on Economics of Peer-to-Peer Systems*, Apr. 2004.

[12] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. Freenet: A distributed anonymous information storage and retrieval system. In *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, 2000.

[13] R. Clayton, S. Murdoch, and R. N. M. Watson. Ignoring the great firewall of China. In *6th Workshop on Privacy Enhancing Technologies*, June 2006.

[14] J. Crandall, D. Zinn, M. Byrd, E. Barr, and R. East. Conceptdoppler: A weather tracker for internet censorship. In *14th ACM Conference on Computer and Communications Security*, 2007.

[15] R. Dingledine. Tor and wikipedia. `http://freehaven.net/~arma/slides-wiki-tor.pdf`.

[16] R. Dingledine and N. Mathewson. Design of a blocking-resistant anonymity system. `http://www.torproject.org/svn/trunk/doc/design-paper/blocking.pdf`.

[17] R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium*, Aug. 2004.

[18] J. Douceur. The sybil attack. In *IPTPS 2002*.

[19] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen. SPKI certificate theory. RFC 2693, Network Working Group, 1986.

[20] N. Feamster, M. Balazinska, G. Harfst, H. Balakrishnan, and D. Karger. Infranet: Circumventing web censorship and surveillance. In *Proceedings of the 11th USENIX Security Symposium*, 2002.

[21] N. Feamster, M. Balazinska, W. Wang, H. Balakrishnan, and D. Karger. Thwarting web censorship with untrusted messenger discovery. In *Privacy Enhancing Technologies Workshop*, Mar. 2003.

[22] D. Liben-Nowell. *An Algorithmic Approach to Social Networks*. PhD thesis, Massachusetts Institute of Technology, June 2005.

[23] S. Marti, P. Ganesan, and H. Garcia-Molina. DHT routing using social links. In *3rd Workshop on Peer-to-Peer Systems (IPTPS 2004)*.

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

[25] M. K. Reiter and A. D. Rubin. Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security*, 1998.

[26] M. T. S. Sepandar D. Kamvar and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the World World Web Conferences (WWW)*, 2003.

[27] N. Y. Times. Thailand bans youtube. `http://www.nytimes.com/2007/04/05/business/worldbusiness/05tube.html`.

[28] R. Toivonen, J.-P. Onnela, J. Saramäki, J. Hyvönen, and K. Kaski. A model for social networks. *Physica A Statistical Mechanics and its Applications*, 371:851–860, 2006.

[29] M. Waldman, A. Rubin, and L. Cranor. Publius: A robust, tamper-evident, censorship-resistant and source-anonymous web publishing system. In *Proceedings of the 9th USENIX Security Symposium*, Aug. 2000.

[30] K. Walsh and E. G. Sirer. Experience with an object reputation system for peer-to-peer filesharing. In *Proceedings of Networked System Design and Implementation (NSDI)*, May 2006.

[31] M. Yang, H. Chen, B. Y. Zhao, Y. Dai, and Z. Zhang. Deployment of a large-scale peer-to-peer social network. In *USENIX WORLDS*, 2004.

[32] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. Defending against sybil attacks via social networks. In *Proceedings of ACM SIGCOMM Conference*, Sept. 2006.

[33] P. Zimmermann. PGP: Source code and internals. 1995.