How to Disappear Completely: A Survey of Private Peer-to-Peer Networks

Michael Rogers[1] and Saleem Bhatti[2]

[1] University College London, London WC1E 6BT, UK
m.rogers@cs.ucl.ac.uk
[2] University of St. Andrews, Fife KY16 9SS, UK
saleem@cs.st-andrews.ac.uk

Abstract. This paper offers a survey of the emerging field of private peer-to-peer networks, which can be defined as internet overlays in which the resources and infrastructure are provided by the users, and new users may only join by personal invitation. The last few years have seen rapid developments in this field, many of which have not previously been described in the research literature. We describe deployed systems, classify them architecturally, and identify some technical and social tradeoffs in the design of private peer-to-peer networks.

1 Introduction

Most peer-to-peer networks are designed to be open to the public: anyone can join a BitTorrent swarm or share files in Gnutella, simply by obtaining the addresses of other participants [1, 2]. Even systems designed to protect the privacy of their users often have open membership policies [3–5]. This openness enables wide participation, ensuring that a large amount of content is available in file sharing networks, for example, but it also allows attackers to monitor, join, and disrupt peer-to-peer networks [6, 7].

In recent years, pervasive surveillance and censorship of the internet and highly publicised lawsuits against users of file sharing networks have led to increasing interest in private, authenticated sharing between groups of friends [8–11]. In a parallel development, creators of collaborative software have sought to combine the flexibility and autonomy of peer-to-peer networks with the confidentiality and authentication provided by traditional groupware [12].

The emergence of private peer-to-peer networks has so far received relatively little attention from the research community – to the best of our knowledge, this paper offers the first survey of developments in this area since the seminal article by Biddle *et al.* more than four years ago [11]. Consequently, many of the references in our survey are to websites rather than to peer-reviewed papers.

The next section defines the scope of this paper and describes some of the technical challenges faced by private peer-to-peer networks. Section 3 provides a survey of deployed systems, and in section 4 we classify the systems architecturally and discuss design tradeoffs. Section 5 concludes the paper.

2 Background

2.1 Definitions

We define a *private peer-to-peer network* as an internet overlay in which the resources and infrastructure are provided by the users, and new users may only join the network by personal invitation. This definition excludes systems that rely on public servers, such as many online social networks and media sharing websites, but it does not necessarily imply decentralisation – some private peer-to-peer networks use central servers, but access to those servers is restricted to invited users, and the servers are owned and operated by users of the network.

Some private peer-to-peer networks allow direct connections between any pair of users, while others only allow direct connections between users who know one another. We will refer to the former as *group-based networks* and the latter as *friend-to-friend networks* [13, 14].

2.2 Technical challenges

Firewalls and network address translators create significant problems for peer-to-peer networks, whether public or private. As the number of internet-connected devices increases, a growing proportion of internet users are behind ‘middleboxes’ of one kind or another. Unfortunately, techniques for establishing connections across middleboxes often require communication with a third party, which can have implications for privacy and autonomy [15–17].

Confidentiality and authentication are two areas where private peer-to-peer networks have an advantage over public networks. Because the users know one another, it is feasible for them to exchange cryptographic keys out-of-band; private peer-to-peer networks could even be bootstrapped using existing keys and trust relationships, such as those embodied in the PGP web of trust [18].

Any system in which the infrastructure is provided collectively by the users faces the problem of encouraging users to contribute resources as well as consuming them [19]. This ’free riding’ problem has been the focus of a great deal of research in public peer-to-peer networks [20–23]. It may be safe to assume that small networks of trusted users will not suffer from free riding to the same extent as public networks, but resource contribution could be an issue for some of the larger private networks, especially those that support indirect anonymous communication.

3 Survey of deployed systems

3.1 Group-based networks

Groove [24, 25] is a groupware application for creating ‘shared spaces’ that can span organisational boundaries. Each member of the group maintains a copy of the shared space’s state, and encrypted updates are transmitted to other members when the state changes. It is not necessary to maintain connections between every pair of members, and indeed firewalls may make this impossible; members who are unable to communicate directly can exchange messages through dedicated relays. Changes to the shared space can be made while members are offline, and synchronised when they reconnect.

Two kinds of shared space can be created. In a mutually trusting space, all changes to the state are authenticated using a single key. This makes it possible for members to spoof updates from other members. In a mutually suspicious space an authentication key is generated for each pair of members, preventing spoofing but increasing the size of update messages.

Members can only join groups by invitation. The inviter is responsible for communicating the new member's encryption and authentication keys to the group – a man-in-the-middle attack is possible at this point, so messages from new members can be spoofed even in mutually suspicious spaces. Any member can evict any other member from a group, which is done by creating a new group key and transmitting it to all members except the evicted member.

Shinkuro [26] and PowerFolder [27] allow decentralised group-based file sharing on local area networks; wide area connections require at least one member of the group to act as a relay. Each group is associated with a shared directory, and changes are synchronised automatically. Direct Connect [28, 29] requires one user to run a server even on local area networks. The server is used for address discovery, keyword searches and chat.

Octopod [30] avoids the need for central servers by using a public distributed hash table for address discovery [31]. Each group is associated with a shared directory, and the group owner can grant read-only or read-write access to other users by sending them the appropriate keys.

WASTE [32, 33] is a group-based network created by Justin Frankel, the author of Gnutella. Like Gnutella it supports flooded queries and reverse path forwarded replies; these are used to implement keyword searches, file sharing and chat. Nodes can relay one another's messages if all-to-all connectivity is not possible. Links are encrypted and optionally padded to a constant traffic level, but there is no end-to-end encryption or authentication, so users can eavesdrop on one another and spoof messages. There are no group keys, so users can only be evicted by removing their keys from every node.

3.2 Friend-to-friend networks

Turtle [34, 35] is a friend-to-friend file sharing network designed for censorship resistance. Searches are flooded through the network, search results are forwarded back along the reverse path, and virtual circuits can be established for anonymous file transfer. The virtual circuit architecture is also capable of supporting other applications, including real-time communication.

Turtle uses a novel key agreement protocol in which friends exchange personal questions, the answers to which are assumed to be known to both users but not to eavesdroppers. This avoids the need for out-of-band key exchange, but the strength of the resulting keys will depend on the extent of the eavesdropper's knowledge about the users.

Version 0.7 of Freenet is also a friend-to-friend network, which prevents attackers from harvesting the addresses of Freenet nodes [36, 37]. This requires a new routing algorithm, since the previous algorithm depended on nodes learning addresses from successful queries [3].

The new algorithm implements a distributed hash table with a circular key space. Whereas conventional distributed hash tables create connections between nodes to produce the required topology, Freenet uses a stochastic algorithm to assign suitable locations to nodes using only the existing connections between friends [38]. All files are stored in the distributed hash table, allowing publishers and readers to remain anonymous.

Freenet stores two kinds of data: content hash keys (CHKs), which are blocks of data identified by their hashes, and signed subspace keys (SSKs), which are blocks of data signed with a private key and identified by the hash of the corresponding public key. SSKs can be updated by anyone who knows the private key and retrieved by anyone who knows the public key, which makes it possible to implement a wide range of services over Freenet, including web browsing, message boards and email. An SSK keypair can be derived from a keyword, in which case anyone who knows the keyword can retrieve and update the SSK.

Like Freenet, GNUnet [39] can be configured to connect only to trusted nodes. GNUnet provides content-based and updatable keys and supports keyword searches [40]. Queries are routed using randomised flooding, which might seem to reveal less information about users than Freenet's social network-based routing. However, Kugler [41] describes a statistical attack that makes it possible, given a long series of related requests, to determine whether the requests are likely to have originated at a neighbouring node or to have been forwarded on behalf of another node. Similar attacks might be possible against Freenet even though its routing algorithm is deterministic, because related requests are routed independently.

SockeToome [42] enables friend-to-friend file transfers between users with dynamic IP addresses, but it is arguably not a peer-to-peer network since no overlay is constructed. In anoNet [43], encrypted VPN tunnels between friends are connected to form a multi-hop overlay. The overlay uses standard internet protocols such as BGP, and even has an internal DNS hierarchy. A reserved network prefix is used to avoid accidentally routing packets between the overlay and the public internet.

Easter [44] uses email as a substrate for friend-to-friend file sharing. This makes it possible to circumvent many firewalls, but the protocol requires frequent polling of email accounts, which might attract the attention of system administrators.

CSpace [45] is a general-purpose friend-to-friend connection service based on a distributed hash table [46]. The connections established by CSpace can be used for any application: file sharing, screen sharing and chat have been implemented so far.

Retroshare [47] is an instant messaging and file sharing network that uses a distributed hash table for address discovery. Users can communicate indirectly through mutual friends and request direct connections. Galet [48], Alliance [49], and Cryptic6 [50] allow friends of friends to communicate in a similar way, but they do not use distributed hash tables or relay servers for address discovery, so addresses and encryption keys must be exchanged out-of-band or through mutually trusted friends.

3.3 Other networks

Sneakernet [51] uses small data-carrying devices such as mobile phones and memory sticks to pass information between friends. A gossip-based protocol allows encrypted messages to travel over multiple hops between a trusted server and anonymous users. Because it relies on a trusted public server, we do not consider Sneakernet to be a private network in the sense defined in section 2.

Many other systems use websites or other public servers to coordinate peer-to-peer communication between friends or in private groups. Recently some client-server instant messaging networks have also begun to support peer-to-peer voice and video connections. We consider such systems to be outside the scope of this survey because of their reliance on public servers.

4 Architecture

The private peer-to-peer networks described in the previous section can be classified architecturally along three axes: scale, visibility, and centralisation.

1. Scale – does the system consist of isolated local networks or a single global network?

The issue of scale raises difficult technical and social tradeoffs. Large private networks are likely to face many of the same connectivity challenges as public peer-to-peer networks, including heterogeneity, churn, and diurnal activity cycles [52, 53]. Because of their size, they are also more likely to attract the attention of eavesdroppers and attackers. Users may feel less of an obligation to contribute to strangers than friends, so free riding may also be an issue for large networks. On the other hand, the wider range of people and resources can make large networks more attractive to potential users [54].

Deployed systems deal with these tradeoffs in a variety of ways. At one end of the axis is WASTE, which is designed for small groups of mutually trusting users; users can belong to more than one network, but traffic does not pass between networks. The group size is limited in practice by the protocol's use of flooding and the difficulty of evicting misbehaving users, which encourages users to be cautious about giving invitations.

At the other end of the axis is Freenet 0.7, which is designed to be a ”globally scalable darknet” [55]. Freenet's routing algorithm is based on the assumption that all users belong to a single small-world social network [38], and it may not be possible to merge mature networks without seriously disrupting routing. The ability to merge networks could be important for growth, because it may be easier for a potential user to find friends who are interested in setting up a local network than to contact and befriend a member of an existing network.

GNUnet and Turtle take an intermediate approach: messages can be forwarded anonymously across the friend-to-friend overlay, so local networks created by small groups of users can be merged by establishing friend-to-friend connections between them. However, GNUnet and Turtle both discover routes by flooding, which does not perform well in large networks; thus even in a merged network, communication may effectively be confined to local regions of the overlay. Indirect communication in Galet, Alliance and Cryptic6 is explicitly local: friends of friends can communicate pseudonymously rather than anonymously, which may allow users to build up trust in new friends before connecting to them directly.

One important insight of Biddle *et al.* is that even when networks are technologically isolated they are socially connected, because users often belong to more than one network. Thus it is possible to speak of ”the darknet” in terms of a patchwork of local ”darknets”, even if there is no single network with global scale.

2. Visibility – can users connect to everyone in the network, or only to their friends? Who else can see that they are participating?

The issue of visibility separates group-based networks from friend-to-friend networks. This distinction becomes more important as networks grow, because any user may invite a friend who does not know all the other users. In a group-based network, the newly invited user will be able to connect to any existing user; thus in terms of privacy, group-based networks become more public as they grow, whereas friend-to-friend networks can (at least in theory) remain private at any scale. Indirect communication through mutual friends, as implemented by Galet, Alliance and Retroshare, represents an intermediate position between group-based and strictly friend-to-friend visibility.

Group-based networks could be vulnerable to Sybil attacks [56], where an attacker uses multiple identities simultaneously, and whitewashing [57], where an attacker changes identities to escape the consequences of past behaviour. For example, it is easy to imagine an attacker automatically ‘inviting’ new identities into a group more quickly than the other users can manually evict them. Friend-to-friend connections are not a panacea for identity-related attacks, but it might be possible to use the structure of social networks to limit the impact of Sybil attacks [58–60].

Regardless of whether the network is group-based or friend-to-friend, users may need to make additional connections to discover the addresses of other users. For example, some networks use public distributed hash tables for address discovery, while others use external servers for NAT and firewall traversal [16]. If participants in a public network can identify the users of a private network, and perhaps even observe which users connect to which others, then many of the benefits of using a private network will have been lost.

It may be easier for group-based networks to avoid relying on external services for address discovery, because only one member of the group needs to have a stable address; the addresses of other members can be learned from that member. In a friend-to-friend network, every user must have one friend with a stable address. Indirect communication through mutual friends could help to alleviate this problem.

3. Centralisation – does the network rely on a central server?

The conventional wisdom is that centralised peer-to-peer networks are fragile, and indeed a number of networks have been successfully shut down by attacking their central servers [10]. However, the risks may be different in private networks, where servers can be more or less hidden from untrusted parties. Centralisation can make it easier to manage identities, exchange cryptographic keys and learn the current addresses of other users, provided all users trust the central server.

Name	Scale	Visibility	Centralisation
Direct Connect	Local	Group	Central server
Groove	Local	Group	Dedicated relays
PowerFolder	Local	Group	Members may act as relays
Shinkuro	Local	Group	Members may act as relays
WASTE	Local	Group	Members may act as relays
Easter	Local	Friends	Email servers
Alliance	Flexible	Friends of friends	Decentralised
Cryptic6	Flexible	Friends of friends	Decentralised
Galet	Flexible	Friends of friends	Decentralised
GNUnet	Flexible	Friends	Decentralised
Turtle	Flexible	Friends	Decentralised
anoNet	Global	Friends	Decentralised
Freenet 0.7	Global	Friends	Decentralised
CSpace	Global	Friends, DHT	Decentralised (DHT)
Retroshare	Global	Friends of friends, DHT	Decentralised (DHT)
Octopod	Global	Group, DHT	Decentralised (DHT)

Table 1. The architecture of deployed private peer-to-peer networks

Direct Connect requires a central server or 'hub' for every group. Many other group-based networks can operate without servers if all users are on the same local area network, but require at least one member to act as a relay for wide area communication. Groove uses dedicated relays that can see who is communicating but cannot decrypt the messages they forward. Easter relies on email servers, which are decentralised but dependent on the centralised domain name system. Most other friend-to-friend systems rely on manual port forwarding or hole punching to traverse NATs and firewalls [15–17]. Friend-to-friend connections can be lost if both friends change their addresses at the same time, and it may be necessary to exchange updated addresses through an alternative channel such as email.

Octopod, Retroshare and CSpace use distributed hash tables for address discovery, which may allow untrusted parties to observe which users connect to which others. Freenet avoids this problem because its DHT implementation only uses existing friend-to-friend connections; users can publish their encrypted contact details under updatable keys for their friends to retrieve anonymously.

5 Conclusions

This paper has provided a brief survey of the emerging field of private peer-to-peer networks, which attempt to combine the flexibility and autonomy of peer-to-peer architectures with the confidentiality and authentication of traditional groupware. Deployed systems can be classified along three architectural axes: scale (local, flexible or global); visibility (group-based or friend-to-friend); and centralisation. Each of these axes involves tradeoffs affecting the robustness, scalability, privacy and ease of use that the resulting systems can provide.

Private peer-to-peer networks are already being used in fields as diverse as business collaboration, secure file sharing, social networking, grassroots political activity, and censorship-resistant communication. Considering the varied and sometimes conflicting requirements raised by these applications, we do not expect that any single network will be able to meet the needs of all users; instead we will continue to see a range of architectures that are adapted to particular uses.

References

1. J. Risson and T. Moors. Survey of research towards robust peer-to-peer networks: Search methods. Technical Report UNSW-EE-P2P-1-1, University of New South Wales, September 2004.
2. E. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim. A survey and comparison of peer-to-peer overlay network schemes. *IEEE Communications Surveys and Tutorials*, 7(2), 2005.
3. I. Clarke, O. Sandberg, B. Wiley, and T.W. Hong. Freenet: A distributed anonymous information storage and retrieval system. In H. Federrath, editor, *Proceedings of the International Workshop on Design Issues in Anonymity and Unobservability, Berkeley, CA, USA*, volume 2009 of *Lecture Notes in Computer Science*, pages 46–66. Springer, 2001.
4. I2P website, `http://www.i2p.net/`.
5. MUTE website, `http://mute-net.sourceforge.net/`.
6. A. Veiga. Music labels tap downloading networks, November 2003. Associated Press news article, available from `http://www.usatoday.com/tech/webguide/music/2003-11-14-sharestats_x.htm`.
7. N. Christin, A.S. Weigend, and J. Chuang. Content availability, pollution and poisoning in file sharing peer-to-peer networks. In *ACM Conference on Electronic Commerce, Vancouver, Canada*, June 2005.
8. J. Pain, editor. *Internet Annual Report*. Reporters Without Borders, 2006.
9. N. Anderson. House approves warrantless wiretapping, September 2006. Ars Technica news article, available from `http://arstechnica.com/news.ars/post/20060929-7867.html`.
10. Electronic Frontier Foundation. RIAA v. the people: Two years later, September 2005. Available from `http://www.eff.org/IP/P2P/RIAAatTWO_FINAL.pdf`.
11. P. Biddle, P. England, M. Peinado, and B. Willman. The darknet and the future of content protection. In *Proceedings of the 2nd International Workshop on Digital Rights Management (DRM 2002), Washington, DC, USA*, volume 2696 of *Lecture Notes in Computer Science*, pages 155–176. Springer, 2003.
12. T. Strufe and D. Reschke. Efficient content distribution in semi-decentralized peer-to-peer networks. In *Proceedings of the 8th International Netties Conference, Ilmenau, Germany*, pages 33–38, September-October 2002.
13. D. Bricklin. Friend-to-friend networks, August 2000. Available from `http://www.bricklin.com/f2f.htm`.
14. L. Gonze. Friendnet, December 2002. Available from `http://www.oreillynet.com/pub/wlg/2428`.
15. B. Ford, P. Srisuresh, and D. Kegel. Peer-to-peer communication across network address translators. In *USENIX Annual Technical Conference, Anaheim, CA, USA*, April 2005.
16. J. Rosenberg, J. Weinberger, C. Huitema, and R. Mahy. RFC 3489: STUN - simple traversal of user datagram protocol (UDP) through network address translators (NATs), March 2003.
17. S. Guha and P. Francis. Characterization and measurement of TCP traversal through NATs and firewalls. In *Internet Measurement Conference (IMC 2005), Berkeley, CA, USA*, October 2005.

18. J. Cederlöf. Web of trust statistics and pathfinder. Available from `http://www.lysator.liu.se/~jc/wotsap/`.
19. R.M. Dawes. Social dilemmas. *Annual Review of Psychology*, 31:169–193, January 1980.
20. E. Adar and B. Huberman. Free riding on Gnutella. *First Monday*, 5(10), October 2000.
21. B. Cohen. Incentives build robustness in BitTorrent. In *Workshop on Economics of Peer-to-Peer Systems, Berkeley, CA, USA*, June 2003.
22. R. Krishnan, M.D. Smith, Z. Tang, and R. Telang. The impact of free-riding on peer-to-peer networks. In *Proceedings of the 37th Hawaii International Conference on System Sciences, Big Island, HW, USA*, pages 199–208, January 2004.
23. S. Nielson, S. Crosby, and D. Wallach. A taxonomy of rational attacks. In M. Castro and R. Renesse, editors, *Proceedings of the 4th International Workshop on Peer-to-Peer Systems (IPTPS '05), Ithaca, NY, USA*, volume 3640 of *Lecture Notes in Computer Science*, pages 36–46. Springer, 2005.
24. J. Udell, N. Asthagiri, and W. Tuvell. Security. In A. Oram, editor, *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*, chapter 18. O'Reilly, March 2001. This chapter describes Groove.
25. Groove Networks website, `http://www.groove.net/`.
26. Shinkuro website, `http://shinkuro.com/`.
27. PowerFolder website, `http://www.powerfolder.com/`.
28. NeoModus Direct Connect website, June 2005, available from `http://web.archive.org/web/20050627012020/www.neo-modus.com/`.
29. DC++ website, `http://dcpp.net/`.
30. Octopod website, `http://sysnet.ucsd.edu/octopod/`.
31. S. Rhea, B. Godfrey, B. Karp, J. Kubiatowicz, S. Ratnasamy, S. Shenker, I. Stoica, and H. Yu. OpenDHT: A public DHT service and its uses. In *SIGCOMM 2005, Philadelphia, PA, USA*, August 2005.
32. WASTE website, `http://waste.sourceforge.net/`.
33. M. Ek, F. Hultin, and J. Lindblom. WASTE peer-to-peer protocol, March 2005. Reverse-engineered protocol documentation, available from `http://prdownloads.sourceforge.net/j-waste/waste_documentation-1.1.pdf?download`.
34. B.C. Popescu, B. Crispo, and A.S. Tanenbaum. Safe and private data sharing with Turtle: Friends team-up and beat the system. In *12th International Workshop on Security Protocols, Cambridge, UK*, April 2004.
35. P. Matějka. Security in peer-to-peer networks. Master's thesis, Department of Software Engineering, Charles University, Prague, December 2004.
36. I. Clarke and O. Sandberg. Routing in the dark: Scalable searches in dark P2P networks. In *DefCon 13, Las Vegas, NV, USA*, July 2005.
37. I. Clarke. A distributed decentralised information storage and retrieval system. Technical report, Division of Informatics, University of Edinburgh, 1999. Available from `http://freenetproject.org/papers/ddisrs.pdf`.
38. O. Sandberg. Distributed routing in small-world networks. In *8th Workshop on Algorithm Engineering and Experiments (ALENEX06), Miami, FL, USA*, January 2006.
39. K. Bennett and C. Grothoff. GAP - practical anonymous networking. In R. Dingledine, editor, *Proceedings of the 3rd International Workshop on Privacy Enhancing Technologies (PET 2003), Dresden, Germany*, volume 2760 of *Lecture Notes in Computer Science*, pages 141–160. Springer, 2003.
40. C. Grothoff, K. Grothoff, T. Horozov, and J.T. Lindgren. An encoding for censorship-resistant sharing, 2005. GNUnet white paper, available from `http://gnunet.org/download/ecrs.ps`.

41. D. Kugler. An analysis of GNUnet and the implications for anonymous, censorship-resistant networks. In R. Dingledine, editor, *Proceedings of the 3rd International Workshop on Privacy Enhancing Technologies (PET 2003), Dresden, Germany*, volume 2760 of *Lecture Notes in Computer Science*, pages 161–176. Springer, 2003.
42. SockeToome website, `http://www.ziggy.speedhost.com/bdsock.html`.
43. anoNet website, `http://anonet.org/`.
44. Easter website, `http://easta.sourceforge.net/`.
45. CSpace website, `http://www.cspace.in/`.
46. P. Maymounkov and David Mazières. Kademlia: A peer-to-peer information system based on the XOR metric. In P. Druschel, F. Kaashoek, and A. Rowstron, editors, *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02), Cambridge, MA, USA*, volume 2429 of *Lecture Notes in Computer Science*, pages 53–65. Springer, 2002.
47. Retroshare website, `http://retroshare.sourceforge.net/`.
48. Galet website, `http://galet.sourceforge.net/`.
49. Alliance website, `http://www.alliancep2p.com/`.
50. Cryptic6 website, `http://cryptic6.sourceforge.net/`.
51. Sneakernet website, `http://informationwithoutborders.endymion.com/`.
52. S. Saroiu, P. Krishna Gummadi, and S.D. Gribble. A measurement study of peer-to-peer file sharing systems. In *Multimedia Computing and Networking (MMCN '02)*, January 2002.
53. D. Stutzbach and R. Rejaie. Towards a better understanding of churn in peer-to-peer networks. Technical Report UO-CIS-TR-04-06, Department of Computer Science, University of Oregon, November 2004.
54. S.J. Liebowitz and S.E. Margolis. Network externalities (effects). In *In New Palgrave Dictionary of Economics and the Law, MacMillan*, 1998.
55. I. Clarke. Project status update, and request for your help, September 2005. Freenet Project announcement, available from `http://emu.freenetproject.org/pipermail/announce/2005-September/000012.html`.
56. J.R. Douceur. The Sybil attack. In P. Druschel, F. Kaashoek, and A. Rowstron, editors, *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02), Cambridge, MA, USA*, volume 2429 of *Lecture Notes in Computer Science*, pages 251–260. Springer, 2002.
57. E. Friedman and P. Resnick. The social cost of cheap pseudonyms. *Journal of Economics and Management Strategy*, 10(2):173–199, 2001.
58. A. Cheng and E. Friedman. Sybilproof reputation mechanisms. In *Proceedings of the 3rd Workshop on Economics of Peer-to-Peer Systems, Philadelphia, PA, USA*, pages 128–132, 2005.
59. J.M. Seigneur, A. Gray, and C.D. Jensen. Trust transfer: Encouraging self-recommendations without Sybil attack. In *Proceedings of the 3rd International Conference on Trust Management (iTrust 2005), Paris, France*, volume 3477 of *Lecture Notes in Computer Science*. Springer, 2005.
60. H. Yu, M. Kaminsky, P.B. Gibbons, and A. Flaxman. SybilGuard: Defending against Sybil attacks via social networks. In *SIGCOMM 2006, Pisa, Italy*, September 2006.