Trust-Based Privacy Preservation for Peer-to-peer Data Sharing *

Yi Lu yilu@cs.purdue.edu
Weichao Wang wangwc@cs.purdue.edu
Dongyan Xu dxu@cs.purdue.edu
Bharat Bhargava bb@cs.purdue.edu

Department of Computer Sciences
Purdue University, West Lafayette, IN 47907

Abstract

Privacy preservation in a peer-to-peer system tries to hide the association between the identity of a participant and the data that it is interested in. We propose a trust-based privacy preservation method for peer-to-peer data sharing. It adopts the trust relation between a peer and its collaborators (buddies). The buddy works as a proxy to send the request and acquire the data. This provides a shield under which the identity of the requester and the accessed data cannot be linked. A privacy measuring method is presented to evaluate the proposed mechanism. Dynamic trust assessment and the enhancement to supplier's privacy are discussed.

1 Introduction

Privacy is information about identifiable persons. In peer-to-peer multimedia streaming systems, it includes identity of peers, content, and interests. Due to security concerns and a need to protect from overload, the requesters and the suppliers keep a certain level of privacy. The increasing amount of data sharing and collaboration calls for privacy-preserving mechanisms. Existing research efforts have studied the anonymous communication problem by hiding the identity of the subject in a group of participants. The proposed schemes ensure that the source of a communication is unknown, but the participants may know the content. In a regulated peer-to-peer community where peer identities are known, the privacy is preserved if a peer's *interest* in some specific data is not revealed. However, if a peer will serve as a supplier for these data after receiving them, privacy is considered to be violated. We investigate the privacy preservation problem by removing the association between the content of the communication and the identity of the source. This is different from assuring anonymity, when identities must not be revealed. Somebody may know the source while others may know the content, but nobody knows both. The approaches will use trusted proxies to protect privacy in a dynamic communication environment.

Earlier research introduces trust [1, 2, 3, 4] into the privacy-preservation mechanisms. Every peer in the community establishes trust relationships with some other peers ('buddies'). The buddies work as 'proxies' during data requesting and streaming. A requester can ask one of its buddies to send out a request on its behalf. Data is streamed from the suppliers to the buddy, and then forwarded to the requester as shown in figure 1. The buddy can also become a supplier in any subsequent session, thus masking the identity of the requester. The privacy of a requester is therefore preserved based on its trust in its buddies.

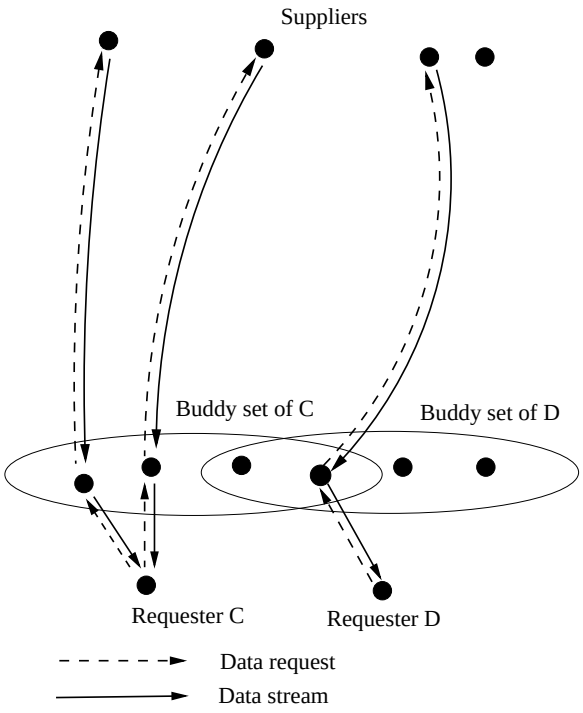

Figure 1: Trust-based privacy preservation

An implementation that adopts static buddy relationships among peers is not adaptable. The requirement for a fully trustworthy buddy limits the number of proxies that a peer can have. It jeopardizes the efforts to hide the requester.

To preserve privacy in a dynamic environment while using trust-based approaches, the following research questions need investigation: How to establish trusted buddy relationships among peers? How to dynamically adjust the trustworthiness of a buddy based on its behavior? How to measure the level of privacy that a spe-

*This research is supported by NSF grants ANI-0219110 and IIS-0209059.

cific approach can achieve? What are the tradeoffs for achieving a certain level of privacy in a peer-to-peer system? How do data sharing and re-distribution policies impact the privacy of a peer? How can the privacy of suppliers be protected? Answers to these questions will provide guidelines for the design of privacy preserving mechanisms for many distributed systems.

The remainder of this paper is organized as follows: In section 2, we review the previous work. Section 3 presents the privacy measuring mechanism. In section 4, the details of the trust-based privacy preservation methods are described. Section 5 and 6 discuss the problems of dynamic trust and experimental studies. Section 7 concludes the paper.

2 Related work

As the amount and variety of data containing user-specific information grows, hiding the suppliers and requesters of data leads to research problems in privacy preservation and anonymity. The existing approaches provide a background for the proposed research.

If the identity of the subject cannot be distinguished from the other $k-1$ subjects, a certain level of anonymity is granted by this uncertainty. The approaches adopting this idea include the k-anonymity [5, 6] and the solutions using multicast or broadcast [7]. In the k-anonymity scheme, the focus is on person-specific data, from which similar subjects can be found with limited efforts. A peer-to-peer system may adopt a similar idea if interest-based clusters can be formed. In solutions such as proxyless MRA [7], the request or data will be sent to a multicast address and may consume too much bandwidth.

Some approaches use fixed servers or proxies to preserve the privacy. Publius [8] protects the identity of a publisher by distributing encrypted data and the k threshold key to a static, system-wide list of servers. However, in a peer-to-peer system, such a server list may not exist. Some anonymity solutions based on trusted third party have been proposed [9]. APFS [7] has been proposed to achieve mutual anonymity in a peer-to-peer file sharing system. Some changes can be adopted so that it can be applied to streaming sessions.

Building a multi-hop path and keeping each node aware of only the previous hop and the next hop has also been used to achieve privacy. The solutions include FreeNet [10, 11], Crowds [12], Onion routing [13, 14], and the shortcut responding protocol [9]. In a peer-to-peer system, a logical neighbor can be far away in terms of network distance. When data streams go through such a multihop path, they may cause a sharp increase in packet loss, delay jitter, and network congestion. These deficiencies can be avoided if a more efficient privacy-preserving solution can be provided.

Research has been conducted on security issues and trust management in peer-to-peer systems [15, 16, 17]. These solutions can be enhanced to provide support for streaming sessions. Results have been done in the area of anonymity [18], location privacy [19, 20], and cooperation among peers [21, 22] in self-organized environments, such as ad hoc networks. They can be tailored and applied to peer-to-peer streaming.

P^5 (Peer-to-Peer Personal Privacy Protocol) [23] provides sender-receiver (supplier-requester) anonymity by transmitting packets to all members of a broadcast group instead of individuals. Every packet is encrypted with the receiver's public key. P^5 achieves scalability by dividing a network into broadcast groups of different sizes. All users are required to generate noise packets so that the amount of traffic is constant at all times. It provides strong anonymity at the cost of efficiency measured in terms of bandwidth utilization. This is because an eavesdropper can not distinguish a data packet from a noise packet.

Herbivore [24] is a peer-to-peer communication system that provides a provable anonymity. It is built on the dining cryptographer networks. Herbivore partitions the network into small anonymizing cliques to address the scalability problem and to decouple the anonymization protocol from the size of the network. It can achieve a high bandwidth and a low latency when deployed over the Internet. The overhead of Herbivore is due to the fact that to anonymously propagate one bit, at least $2(k-1)$ bits are sent, where k is the size of the clique. Whenever a node is sending a packet, to achieve anonymity, all other nodes in the clique have to send at least the same amount of data. This idea may be adjusted for an environment where peers have limited bandwidth.

3 Privacy measurement

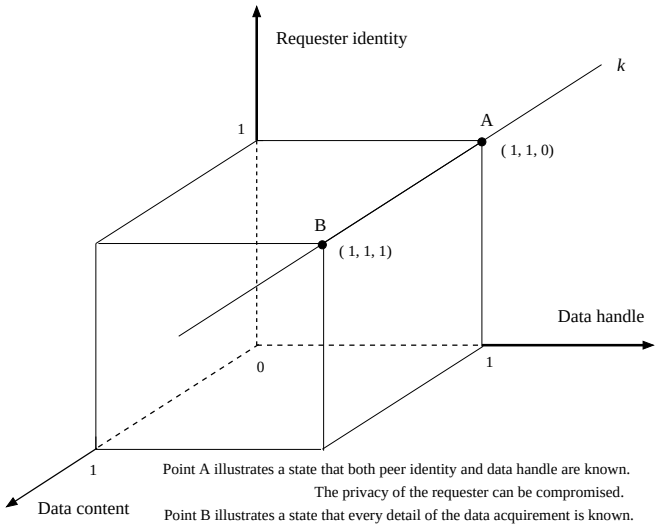

Figure 2: Privacy measurement

A tuple ⟨requester ID, data handle, data content⟩ is defined to describe information that a peer possesses when data is acquired. Figure 2 is a visualization of privacy measurement. Data handle is used to identify the requested data (e.g. file name and the segment index). For each tuple element, '0' means that the peer knows

nothing, while '1' means that it knows everything. For example, a supplier's vector is $\langle$x, 1, 1$\rangle$ (x ϵ [0,1]) because it knows all details of the requested data. A state in which a requester's privacy is compromised can be represented as a vector $\langle$1, 1, y$\rangle$ (y ϵ [0,1]), from which one can link the identity of the requester with data that it is interested in.

An operation "*" is defined as follows: $\langle a_1, a_2, a_3\rangle * \langle b_1, b_2, b_3\rangle = \langle c_1, c_2, c_3\rangle$, where

$$c_i = \begin{cases} \max(a_i, b_i), & a_i \neq 0 \text{ and } b_i \neq 0; \\ 0, & \text{otherwise.} \end{cases}$$

It describes the revealed information after a collusion of two peers when each knows a part of the "secret". For example, a buddy can compromise the privacy of both the supplier and the requester if it knows every detail of a stream, that is, if its vector is $\langle 1, 1, 1\rangle$. On the other hand, if the data handle is encrypted and a buddy does not see the plain text, it has to collude with the supplier to compromise the privacy of the requester. At least one "*" operation is required. This approach has the potential of providing a higher privacy-preservation level.

To measure privacy levels, a weighting function $W()$ if defined. $W(\langle a_1, a_2, a_3\rangle)$ is the effort that is required for a privacy violator to obtain this information. The most important characteristic of $W()$ is

$$W(\langle c_1, c_2, c_3\rangle) \leq W(\langle a_1, a_2, a_3\rangle) + W(\langle b_1, b_2, b_3\rangle)$$

if $\langle c_1, c_2, c_3\rangle = \langle a_1, a_2, a_3\rangle * \langle b_1, b_2, b_3\rangle$

The privacy-preserving level provided by a solution is defined as the effort required to reach a privacy compromising state $W(\langle a_1, a_2, a_3\rangle)$. Usually, $\langle a_1, a_2, a_3\rangle$ is $\langle 1, 1, 1\rangle$.

4 Trust-based privacy preservation schemes

A series of privacy enhancing mechanisms are proposed, which can be built into the system. These mechanisms provide increasing levels of privacy at the cost of computation and communication overhead.

4.1 The idea

The idea of trust-based privacy preservation is illustrated in figure 3. In stead of sending the request by itself, the requester asks one or several of its buddies to look up the data on its behalf (step 1 and 2 in figure 3). Once the supplier is located, the buddy will serve as a proxy to deliver the data to the requester (step 3 and 4). Other peers, including the suppliers, may know that the buddy is requesting something, but they would not know who is really interested in it. The requester's privacy is protected. However, the requester's information is known to the buddy. Its privacy solely depends on the trustworthiness and reliability of the buddy. The

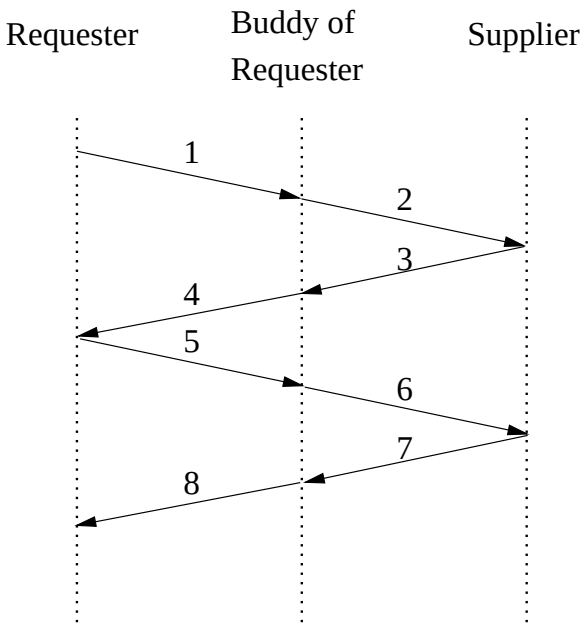

Figure 3: Privacy preservation through the buddy

privacy level can be measured by W_1, which is the effort needed to comprise the buddy.

To improve the achieved privacy level, the data handle is not put in the request at the very beginning. When a requester initiates its request, it calculates the hash value of the handle and reveals only a part of the hash result in the request sent to a buddy (step 1 and 2). Each peer receiving the request compares this revealed partial hash to the hash codes of the data handles that it holds. Depending on the length of the revealed part, the receiving peer may find multiple matches. This does not imply that the peer has the requested data. Thus this peer will provide a candidate set, along with a certificate of its public key, to the requester. If the matched set is not empty, the peer will construct a Bloom filter [25] based on the left parts of the matched hash codes, and send it back to the buddy. The buddy forwards it back to the requester (step 3 and 4). Examining the filters, the requester can eliminate from the candidate supplier list all peers that do not have the required data. It then encrypts the complete request with the supplier's public key and gets the requested data with the help from its buddy (step 5, 6, 7, and 8). Through adjusting the length of the revealed hash code, the requester can control the number of eliminated peers. The privacy level is measured by $W_1 + W_2$, where W_2 is the effort needed to break the bloom filter and hash function.

This mechanism has two advantages: (a) It is difficult to infer a data handle from a partial hash result, unless an adversary conducts a brute force attack on all existing data handles. (b) For the peers that have the required data, the requester can adjust the length of the revealed hash code to partially hide what it wants. False hits are possible when a peer does not have data but the Bloom filter shows that it might have it. The allowable error of

this mechanism can be determined [25].

Although the privacy-preservation level will increase in the look up phase using the above mechanism, the privacy of the requester is still vulnerable if the buddy can see the data content when it relays the data for the requester. So the privacy level of the above scheme is still W_1 in the worst case. To improve privacy assurance and prevent eavesdropping, we can encrypt the data handle and the data content. If the identity of the supplier is known to the requester, it can encrypt the request using the supplier's public key. The public key of the requester cannot be used because the certificate will reveal its identity.

The solution works as follows: The requester generates a symmetric key and encrypts it using a supplier's public key. Only the supplier can recover the key and use it to encrypt data. Thus the data transmission in step 5, 6, 7, and 8 is protected by encryption. To prevent a buddy of the requester from conducting man-in-the-middle attacks, the buddy is required to sign the packet. This provides a non-repudiation evidence, and shows that the packet is not generated by the buddy itself. The privacy level of this scheme is $W_1 + min(W_2, W_3)$, where W_3 is the effort needed to break the encryption.

4.2 Enhancement

The above scheme prevents a single peer from obtaining the entire information about a data sharing request. However, if the buddy of the requester knows who the supplier is, it can collude with the supplier to reveal the interest of the requester. The fact that this supplier has this data will also be revealed, thus the supplier's privacy is violated. In the extreme case, the privacy level of the above scheme is $W_1 + min(W_2, W_3, W_4)$, where W_4 is the effort needed to comprise the supplier.

To address these problems, the proposed mechanism can be improved by having the supplier respond to a request via its own buddies as shown in figure 4. The buddy of the supplier cannot violate the privacy of the supplier, because the request is protected by the hash function and bloom filter and the data is protected by the end-to-end encryption. This will increase the privacy level to $W_1 + min(W_2, W_3, W_4 + W_5)$, where W_5 is the effort needed to comprise the buddy of the supplier.

The adoption of buddies to protect privacy may increase the communication overhead in a peer-to-peer network, because the requester and supplier may be close to each other while the buddies are far away. The impacts can be studied through simulation.

5 Trustworthiness of peers

A model will be built to assess the trustworthiness of a buddy based on its behaviors and other peers' recommendations. The peer's behaviors, such as keeping a secret while being a proxy, forwarding requests in a timely fashion, buffering data to improve streaming-capacity, etc., are all parameters that together affect the trustworthiness metric. Communication principles, such as

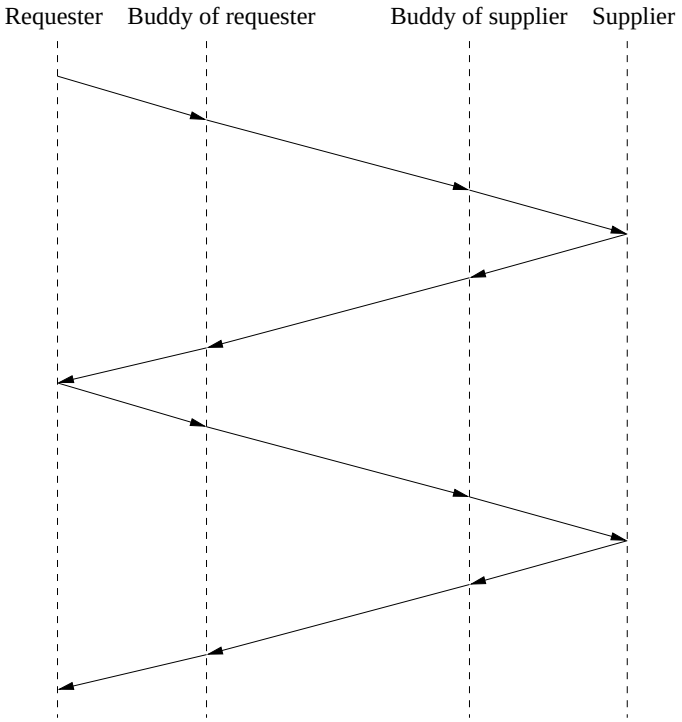

Figure 4: Enhancement

Kalman filtering [26], can be applied to build a trust model as a multivariate, time-varying state vector that utilizes past information to predict future performance. Using this model, the trust in the buddies can be dynamically updated with each fulfilled request.

The assessment of the trustworthiness will be based on our current research on trust formalization. This research will investigate how to collect information from other peers [27], and how trust in their recommendations affects the trustworthiness of a buddy in different privacy related schemes.

6 Experimental study

To conduct extended evaluation on the scalability of the proposed mechanism, a large-scale prototype will be developed and deployed in PlanetLab, a wide-area distributed system testbed. It has been observed that the data distribution capability of the system is dependent on the "buddy" relation among peers. Especially, it is possible that even with a large peer population, the overall capacity is low because of the lack of buddy relations in the system. It is also possible that a small set of highly "friendly" peers (i.e. buddies of many peers) might become overloaded because they are involved in too many data sharing sessions. By performing experiments in a large scale network, more insights can be obtained about the limitations and the bounds of peer-to-peer data sharing capacity under various levels of privacy requirements.

Experiments will be conducted to determine the values of the parameters in the trustworthiness assessment algorithm, and to evaluate the overheads of different combinations of privacy enhancing mechanisms. Prototypes for trust-enhanced role assignment (TERA) [28] and other supporting software products are being developed. We briefly outline one experiment.

Purpose: The trustworthiness value of a buddy, as viewed by a requester, is impacted by both direct experiences of the requester, and the recommendations made by other peers. The purpose of this experiment is to determine the values of the parameters in the trustworthiness assessment algorithm.

Input parameters: Evidence obtained by requester via direct experience in dealing with its buddy, the recommendations for the buddy from other peers, the trust values for the recommenders maintained by the requester, and the history of the trust values for the buddy.

Output parameters: Direct experiences are considered in a fading manner. Output parameters include the length of the remembered history, and the fading factor value. Another output result is the mapping function between trust values for the recommenders and a weight of each recommendation.

Method: The largest change in the trust value that can be caused by a recommendation is predetermined by the peers. The fading speed and the mapping function are calculated recursively. The parameters are determined by the least square error method. When the difference between the predicted trust value and the observed value exceeds a threshold, the algorithm will change the values of the parameters. This makes the algorithm adaptable to the changes in peer's behavior patterns.

Analysis and observation: We consider the trustworthiness requirement as an independent variable, and the procedure to determine the parameters as a cost function. We will identify how the costs are affected by the dynamics of trust values. Observations will also help to explore the robustness of the trust and privacy mechanisms against false recommendations. The result will provide the guidelines for achieving a better efficiency/accuracy tradeoff in trustworthiness assessment.

7 Conclusion

In a peer-to-peer system in which the identities of the participants are known, enforcing privacy is different from the traditional node anonymity problem. In this paper, we propose a trust-based privacy preservation method for peer-to-peer data sharing. It adopts the buddy of a peer as the proxy during the data acquirement. The requester sends the request and gets the data through this proxy, which makes it difficult for the eavesdropers and other peers to explore the real interest of the node. A privacy measuring method is presented to evaluate the proposed mechanism. As an enhancement, the scheme to protect the privacy of the suppliers is also discussed.

The immediate extensions to the proposed work focus on the following aspects: (1) Solid analysis and experiments on large scale networks are required to study the distribution of the buddies and its impacts on data sharing. (2) A security analysis of the proposed mechanism is required. The extensions will provide guidelines for the improvements of the proposed method and lead to a better privacy preservation mechanism for peer-to-peer systems.

References

[1] B. Bhargava and Y. Zhong, "Authorization based on evidence and trust," in *Proc. of International Conference on Data Warehousing and Knowledge Discovery (DaWaK'02)*, Aix-en-Provence, France, September 2002.

[2] B. Bhargava, "Vulnerabilities and fraud in computing systems," in *Proc. of International Conference on Advances in Internet, Processing, Systems, and Interdisciplinary Research (IPSI'03)*, Sveti Stefan, Serbia and Montenegro, October 2003.

[3] L. Lilien and A. Bhargava, "From vulnerabilities to trust: A road to trusted computing," in *Proc. of International Conference on Advances in Internet, Processing, Systems, and Interdisciplinary Research (IPSI'03)*, Sveti Stefan, Serbia and Montenegro, October 2003.

[4] L. Lilien, "Developing pervasive trust paradigm for authentication and authorization," in *Proc. of Third Cracow Grid Workshop (CGW'03)*, Krakow (Cracow), Poland, October 2003.

[5] L. Sweeney, "Achieving k-anonymity privacy protection using generalization and suppression," *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, vol. 10, no. 5, pp. 571–588, 2002.

[6] Sweeney, "K-anonymity: A model for protecting privacy," *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[7] V. Scarlata, B. Levine, and C. Shields, "Responder anonymity and anonymous peer-to-peer file sharing," in *Proc. of IEEE International Conference on Network Protocols (ICNP)*, Riverside, CA, 2001.

[8] M. Waldman, A. D. Rubin, and L. F. Cranor, "Publius: A robust, tamper-evident, censorship-resistant, web publishing system," in *Proc. 9th USENIX Security Symposium*, August 2000, pp. 59–72.

[9] L. Xiao, Z. Xu, and X. Zhang, "Low-cost and reliable mutual anonymity protocols in peer-to-peer networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 14, no. 9, pp. 829–840, 2003.

[10] I. Clarke, O. Sandberg, B. Wiley, and T. Hong, "Freenet: A distributed anonymous information storage and retrieval system," in *Proc. of Workshop on Design Issues in Anonymity and Unobservability*, Berkeley, CA, 2000.

[11] I. Clarke, S. Miller, T. Hong, O. Sandberg, and B. Wiley, "Protecting free expression online with freenet," *IEEE Internet Computing*, pp. 40–49, January 2002.

[12] M. K. Reiter and A. D. Rubin, "Crowds: anonymity for Web transactions," *ACM Transactions on Information and System Security*, vol. 1, no. 1, pp. 66–92, 1998.

[13] D. Goldschlag, M. Reed, and P. Syverson, "Onion routing for anonymous and private internet connections," *Communications of The ACM*, vol. 42, no. 2, 1999.

[14] M. Reed, P. Syverson, and D. Goldschlag, "Anonymous connections and onion routing," *IEEE Journal on Selected Areas in Communication Special Issue on Copyright and Privacy Protection*, 1998.

[15] K. Aberer and Z. Despotovic, "Managing trust in a peer-to-peer information system," in *Proc. of Ninth International Conference on Information and Knowledge Management (CIKM'01)*, Atlanta, GA, November 2001.

[16] J. Bailes and G. Templeton, "Managing p2p security," *Communications of the ACM*, vol. 47, no. 9, pp. 95–98, 2004.

[17] M. Agarwal, "Security issues in p2p systems," www.ece.rutgers.edu/ parashar/Classes/01-02/ece579/slides/security.pdf, 2002.

[18] J. Kong and X. Hong, "ANODR: anonymous on demand routing with untraceable routes for mobile ad-hoc networks," in *Proc. of 4th ACM international symposium on Mobile ad hoc networking and computing*, Annapolis, MD, June 2003.

[19] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica, "Geographic routing without location information," in *Proc. of ACM International Conference on Mobile Computing and Networking*, San Diego, CA, September 2003.

[20] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, "Localization from mere connectivity," in *Proc. of the 4th ACM international symposium on Mobile ad hoc networking and computing*, Annapolis, MD, June 2003.

[21] S. Buchegger and J. L. Boudec, "Cooperation of nodes. in: L. Buttyan and J.-P. Hubaux (eds.), report on a working session on security in wireless ad hoc networks," *ACM Mobile Computing and Communications Review (MC2R)*, vol. 6, no. 4, 2002.

[22] L. Buttyan and J.-P. Hubaux, "Enforcing service availability in mobile ad-hoc WANs," in *Proc. of First IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC)*, Boston, MA, August 2000.

[23] R. Sherwood, B. Bhattacharjee, and A. Srinivasan, "p^5: A protocol for scalable anonymous communication," in *Proc. of IEEE Symposium on Security and Privacy*, Oakland, CA, May 2002, pp. 53 –65.

[24] S. Goel, M. Robson, M. Polte, and E. Sirer, "Herbivore: A scalable and efficient protocol for anonymous communication," Cornell University, CIS Technical Report TR2003-1890, February 2003.

[25] B. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of The ACM*, vol. 13, no. 7, pp. 422–426, July 1970.

[26] R. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME Journal of Basic Engineering*, vol. 8, pp. 35–45, 1960.

[27] N. Li, W. H. Winsborough, and J. C. Mitchell, "Distributed credential chain discovery in trust management," *Journal of Computer Security*, vol. 11, no. 1, pp. 35–86, February 2003.

[28] "Trust-enhanced role assignment (TERA) prototype," *http://raidlab.cs.purdue.edu/zhong/NSFTrust/*.