Special Issue Correspondence

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

Yi Lu, *Member, IEEE*, Weichao Wang, *Member, IEEE*, Bharat Bhargava, *Fellow, IEEE*, and Dongyan Xu, *Member, IEEE*

***Abstract*—Privacy preservation in a peer-to-peer (P2P) system tries to hide the association between the identity of a participant and the data that it is interested in. This paper proposes a trust-based privacy-preservation method for P2P 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.**

***Index Terms*—Peer to peer, privacy preservation, proxy, trust based.**

I. Introduction

Privacy is information about identifiable persons. In peer-to-peer (P2P) multimedia streaming systems, it includes identity of peers, content, and interests. Due to security concerns and a need to protect from an 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 P2P 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]–[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 are streamed from the suppliers to the buddy, and then forwarded to the requester, as shown in Fig. 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.

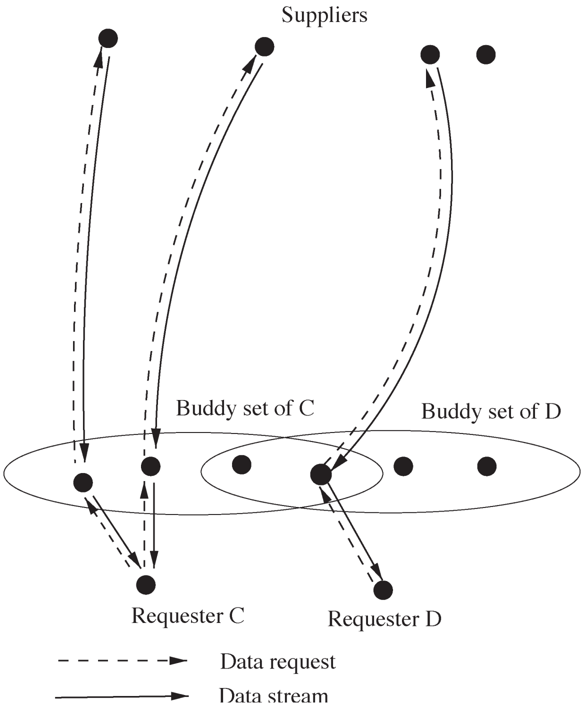

Fig. 1. Data acquirement in trust-based privacy preservation.

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 specific approach can achieve? What are the tradeoffs for achieving a certain level of privacy in a P2P system? How do data-sharing and redistribution 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.

Proxies have been applied to handle the request–supply relationship in the systems such as role-based access control (RBAC) [5], [6]. Special advantages have been brought to P2P systems by the adoption of proxies. First, different from the privacy-preservation methods using k-anonymity, the approach that uses proxies avoids the broadcast or multicast of the contents in the networks, which will save the prestigious bandwidth in pervasive systems. Second, various research has shown that either preestablished or postestablished trust relationship exists in the P2P systems [7], [8]. Using proxy takes advantage of such relationship to improve the safety of the system. Third, the highly dynamic membership in the P2P systems makes the "buddies" volatile, which reduces the probability that a malicious node identifies a user by its proxies.

Manuscript received December 15, 2004; revised September 10, 2005. This work was supported by the National Science Foundation under Grants ANI-0219110, IIS-0209059, and IIS-0242840. This paper was recommended by Guest Editors H. R. Rao and S. J. Upadhyaya.

Y. Lu is with Microsoft Corporation, Redmond, WA 98052 USA (e-mail: yilu@cs.purdue.edu).

W. Wang is with the Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, KS 66045 USA (e-mail: weichaow@eecs.ku.edu).

B. Bhargava and D. Xu are with the Department of Computer Sciences, Purdue University, West Lafayette, IN 47907 USA (e-mail: dxu@cs.purdue.edu; bb@cs.purdue.edu).

Digital Object Identifier 10.1109/TSMCA.2006.871795

1083-4427/$20.00 © 2006 IEEE

The remainder of this paper is organized as follows: In Section II, we review the previous work. Section III presents the privacy measuring mechanism. In Section IV, the details of the trust-based privacy-preservation methods are described. Sections V and VI discuss the problems of dynamic trust and experimental studies. Section VII concludes the paper.

II. Related Work

As the amount and variety of data that contains user-specific information grows, hiding the suppliers and requesters of the 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 [9], [10] and the solutions using multicast or broadcast [11]. In the k-anonymity scheme, the focus is on person-specific data, from which similar subjects can be found with limited efforts. A P2P system may adopt a similar idea if interest-based clusters can be formed. In solutions such as proxyless multicast responder anonymity (MRA) [11], 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 [12] 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 P2P system, such a server list may not exist. Some anonymity solutions based on trusted third party have been proposed [13]. Anonymous peer-to-peer file sharing (APFS) [11] has been proposed to achieve mutual anonymity in a P2P file sharing system. Some changes can be adopted so that it can be applied to streaming sessions.

Building a multihop 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 [14], [15], Crowds [16], Onion routing [17], [18], and the shortcut responding protocol [13]. In a P2P 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 developed.

Research has been conducted on security issues and trust management in P2P systems [8], [19], [20]. These solutions can be enhanced to provide support for streaming sessions. Results have been collected in the areas of anonymity [21], location privacy [22], [23], and cooperation among peers [24], [25] in self-organized environments, such as ad hoc networks. They can be tailored and applied to P2P streaming.

P2P personal privacy protocol (P^5) [26] 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 cannot distinguish a data packet from a noise packet.

Herbivore [27] is a P2P communication system that provides 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.

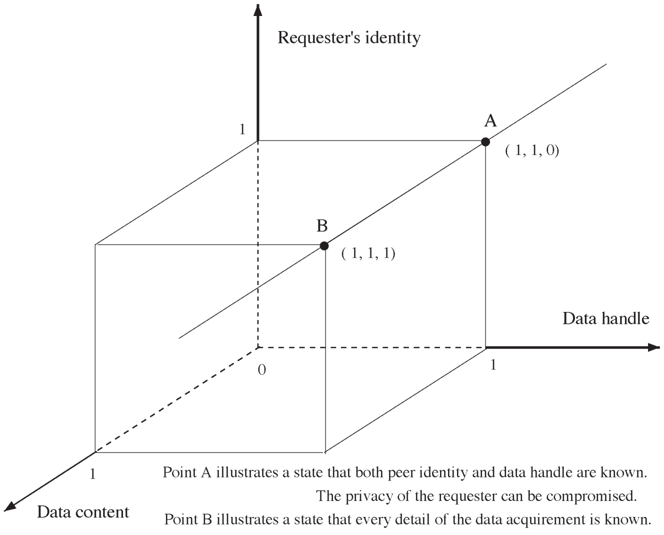
Requester's identity
1
A
(1, 1, 0)
B
(1, 1, 1)
Data handle
0
1
1
Data content
Point A illustrates a state that both peer identity and data handle are known.
The privacy of the requester can be compromised.
Point B illustrates a state that every detail of the data acquirement is known.

Fig. 2. Privacy-measurement method.

III. Privacy Measurement

A tuple ⟨requester's ID, data handle, data content⟩ is defined to describe information that a peer possesses when data are acquired. Fig. 2 is a visualization of the privacy-measurement method. 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 $\langle \mathrm{x}, 1, 1\rangle$ $(\mathrm{x}\epsilon[0,1])$ 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 $\langle 1, 1, \mathrm{y}\rangle$ $(\mathrm{y}\epsilon[0,1])$, 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 level of privacy preservation.

To measure privacy levels, a weighting function $W()$ is defined. $W(\langle a_1, a_2, a_3\rangle)$ describes the efforts that are 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 efforts that are 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 $\langle 1, 1, 1\rangle$.

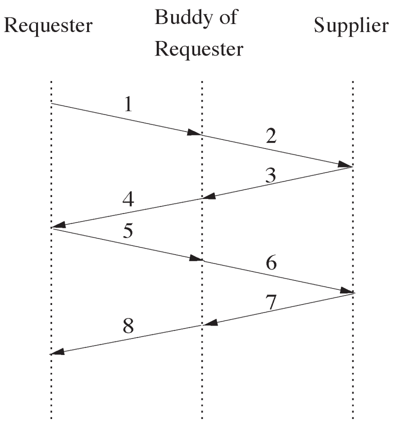

Fig. 3. Data acquirement through the buddy of the requester.

IV. Trust-Based Privacy-Preservation Schemes

In this part, 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.

A. Idea

Protecting the Identity of the Requester: The mechanism to hide the identity of the requester is illustrated in Fig. 3. Instead of sending the request by itself, the requester asks one or several of its buddies to look up the data on its behalf (steps 1 and 2 in Fig. 3). Once the supplier is located, the buddy will serve as a proxy to deliver the data to the requester (steps 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. Since the requester's information is known to the buddy, its privacy solely depends on the trustworthiness and reliability of the buddy. The privacy level can be measured by the efforts that are needed to comprise the buddy.

Protecting the Data Handle: 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 (steps 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 [28] 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 (steps 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 (steps 5–8). Through adjusting the length of the revealed hash code, the requester can control the number of eliminated peers. The privacy level has been improved since the malicious nodes need to both compromise the buddy and break the Bloom filter and hash function.

This mechanism has two advantages: 1) 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. 2) 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 [28].

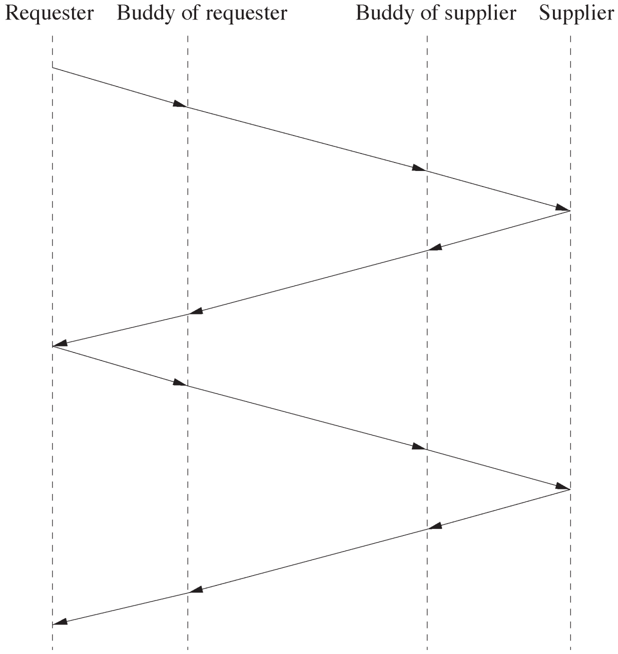

Fig. 4. Data acquirement through the buddies of requester and supplier.

Hiding the Data Content: Although the privacy-preservation level has been improved during the look up phase using the previous schemes, the privacy of the requester is still vulnerable if the buddy can see the data content when it relays the packets for the requester. 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 steps 5–8 are 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 nonrepudiation evidence, and shows that the packet is not generated by the buddy itself. The privacy level has been improved since now the malicious nodes need to break the encryption keys as well.

B. Enhancement

The previous 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.

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 Fig. 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 are protected by the end-to-end encryption.

The adoption of buddies to protect privacy may increase the communication overhead in a P2P 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.

V. Trustworthiness of Peers

A model has been 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 Kalman filtering [29], are 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 notions of trusting belief, disposition to trust, and context are represented in a computational trust model. Trusting belief is a truster's belief in a trustee's competence or integrity. A trusting belief consists of the belief value and predictability, which are real numbers ranging over [0, 1]. Predictability characterizes the goodness of the belief. Disposition to trust is represented as a truster's profiles that consist of prior trusting beliefs and three types of belief formation policies: method preference policies, imprecision handling, and uncertainty handling policies. Two functions have been defined to relate contexts so that trusting belief can be built using cross-context information. The details of the model can be found in [30].

The assessment of the trustworthiness is based on our current research on trust formalization. This research investigates how to collect information from other peers [31], and how trust in their recommendations affects the trustworthiness of a buddy in different privacy-related schemes.

VI. Experimental Study

To conduct extended evaluation on the scalability of the proposed mechanism, a large-scale prototype has been developed. It has been observed that the data distribution capability of the system is dependent on the "buddy" relations among peers. Especially, it is possible that even with a large population of peers, 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 P2P data-sharing capacity under various levels of privacy requirements.

Experiments are 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) [32] 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: Evidences obtained by requester via direct experiences 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 guidelines for achieving a better efficiency/accuracy tradeoff in trustworthiness assessment.

VII. Conclusion

In a P2P 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 P2P 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 eavesdroppers 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) A solid analysis and experiments in 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 to the proposed method and lead to a better privacy-preservation mechanism for P2P systems.

References

[1] B. Bhargava and Y. Zhong, "Authorization based on evidence and trust," in *Proc. Int. Conf. DaWaK*, Aix-en-Provence, France, Sep. 2002, pp. 94–103.

[2] B. Bhargava, "Vulnerabilities and fraud in computing systems," in *Proc. Int. Conf. Advances IPSI Research*, Sveti Stefan, Serbia and Montenegro, Oct. 2003.

[3] L. Lilien and A. Bhargava, "From vulnerabilities to trust: A road to trusted computing," in *Proc. Int. Conf. Advances IPSI Research*, Sveti Stefan, Serbia and Montenegro, Oct. 2003.

[4] L. Lilien, "Developing pervasive trust paradigm for authentication and authorization," in *Proc. 3rd CGW*, Krakow (Cracow), Poland, Oct. 2003, pp. 42–49.

[5] T. Chalfant, "Role based access control and secure shell—A closer look at two solaris operating environment security features," *Sun Microsystems Blueprint*, Jun. 2003.

[6] E. Lupu and M. Sloman, "Reconciling role based management and role based access control," in *Proc. 2nd ACM Workshop Role Based Access Control*, 1997, pp. 135–141.

[7] A. Singh and L. Liu, "Trustme: Anonymous management of trust relationships in decentralized P2P systems," in *Proc. 3rd IEEE Int. Conf. Peer-to-Peer Computing*, 2003, pp. 142–149.

[8] K. Aberer and Z. Despotovic, "Managing trust in a peer-to-peer information system," in *Proc. 9th Int. CIKM*, Atlanta, GA, Nov. 2001, pp. 310–317.

[9] L. Sweeney, "Achieving k-anonymity privacy protection using generalization and suppression," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 571–588, Oct. 2002.

[10] Sweeney, "K anonymity: A model for protecting privacy," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, no. 5, pp. 557–570, Oct. 2002.

[11] V. Scarlata, B. Levine, and C. Shields, "Responder anonymity and anonymous peer-to-peer file sharing," in *Proc. IEEE ICNP*, Riverside, CA, 2001, pp. 272–280.

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

[13] L. Xiao, Z. Xu, and X. Zhang, "Low-cost and reliable mutual anonymity protocols in peer-to-peer networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 14, no. 9, pp. 829–840, Sep. 2003.

[14] I. Clarke, O. Sandberg, B. Wiley, and T. Hong, "Freenet: A distributed anonymous information storage and retrieval system," in *Proc. Workshop Design Issues Anonymity and Unobservability*, Berkeley, CA, 2000, pp. 45–66.

[15] I. Clarke, S. Miller, T. Hong, O. Sandberg, and B. Wiley, "Protecting free expression online with freenet," *IEEE Internet Comput.*, vol. 6, no. 1, pp. 40–49, Jan. 2002.

[16] M. K. Reiter and A. D. Rubin, "Crowds: Anonymity for Web transactions," *ACM Trans. Inf. Syst. Secur.*, vol. 1, no. 1, pp. 66–92, 1998.

[17] D. Goldschlag, M. Reed, and P. Syverson, "Onion routing for anonymous and private internet connections," *Commun. ACM*, vol. 42, no. 2, pp. 39–41, 1999.

[18] M. Reed, P. Syverson, and D. Goldschlag, "Anonymous connections and onion routing," *IEEE J. Sel. Areas Commun.—Special Issue on Copyright and Privacy Protection*, vol. 16, no. 4, pp. 482–494, May 1998.

[19] J. Bailes and G. Templeton, "Managing P2P security," *Commun. ACM*, vol. 47, no. 9, pp. 95–98, Sep. 2004.

[20] M. Agarwal, *Security Issues in P2P Systems*. (2002). [Online]. Available: www.ece.rutgers.edu/ parashar/Classes/01-02/ece579/slides/security.pdf

[21] J. Kong and X. Hong, "ANODR: Anonymous on demand routing with untraceable routes for mobile ad-hoc networks," in *Proc. 4th ACM Int. Symp. Mobile Ad Hoc Networking and Computing*, Annapolis, MD, Jun. 2003, pp. 291–302.

[22] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica, "Geographic routing without location information," in *Proc. ACM Int. Conf. Mobile Computing and Networking*, San Diego, CA, Sep. 2003, pp. 96–108.

[23] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, "Localization from mere connectivity," in *Proc. 4th ACM Int. Symp. Mobile Ad Hoc Networking and Computing*, Annapolis, MD, Jun. 2003, pp. 201–212.

[24] S. Buchegger and J. L. Boudec, "Cooperation of nodes," extended abstract in L. Buttyàn and Hubaux (eds.), "Report on a working session on security in wireless ad hoc networks", *ACM Mobile Comput. Commun. Rev. (MC2R)*, vol. 7, no. 1, pp. 74–94, 2003.

[25] L. Buttyan and J.-P. Hubaux, "Enforcing service availability in mobile ad-hoc WANs," in *Proc. 1st IEEE/ACM MobiHOC*, Boston, MA, Aug. 2000, pp. 87–96.

[26] R. Sherwood, B. Bhattacharjee, and A. Srinivasan, "p^5: A protocol for scalable anonymous communication," in *Proc. IEEE Symp. Security and Privacy*, Oakland, CA, May 2002, pp. 58–70.

[27] S. Goel, M. Robson, M. Polte, and E. Sirer, "Herbivore: A scalable and efficient protocol for anonymous communication," Dept. Comput. Inform. Sci., Cornell Univ., Ithaca, NY, CIS Tech. Rep. TR2003-1890, Feb. 2003.

[28] B. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Commun. ACM*, vol. 13, no. 7, pp. 422–426, Jul. 1970.

[29] R. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME J. Basic Eng.*, ser. D, vol. 82, pp. 35–45, 1960.

[30] Y. Zhong, "Formalization of dynamic trust and uncertain evidence for user authorization," Ph.D dissertation, Dept. Comput. Sci., Purdue Univ., West Lafayette, IN, 2005.

[31] N. Li, W. H. Winsborough, and J. C. Mitchell, "Distributed credential chain discovery in trust management," *J. Comput. Secur.*, vol. 11, no. 1, pp. 35–86, Feb. 2003.

[32] Trust-Enhanced Role Assignment (TERA) Prototype. (2004). http://raidlab.cs.purdue.edu/zhong/NSFTrust/

A Scheme for Privacy-Preserving Data Dissemination

Leszek Lilien, *Senior Member, IEEE*, and
Bharat Bhargava, *Fellow, IEEE*

***Abstract*—An adequate level of trust must be established between prospective partners before an interaction can begin. In asymmetric trust relationships, one of the interacting partners is stronger. The weaker partner can gain a higher level of trust by disclosing private information. Dissemination of sensitive data owned by the weaker partner starts at this moment. The stronger partner can propagate data to others, who may then choose to spread data further. The proposed scheme for privacy-preserving data dissemination enables control of data by their owner (such as a weaker partner). It relies on the ideas of bundling sensitive data with metadata, an apoptosis of endangered bundles, and an adaptive evaporation of bundles in suspect environments. Possible applications include interactions among patients and healthcare providers, customers and businesses, researchers, and suppliers of their raw data. They will contribute to providing privacy guarantees, which are indispensable for the realization of the promise of pervasive computing.**

***Index Terms*—Data dissemination, data privacy protection, privacy, trust.**

I. Introduction

Any interaction, from a simple transaction to a complex collaboration, can start only after an adequate level of trust exists between interacting entities. *Trust* is defined as "reliance on the integrity, ability, or character of a person or thing" [1].

Use of trust is often implicit. Quite frequently, it is gained offline [6]. A user who downloads a file from an unfamiliar Web site trusts it implicitly by not even considering trust in a conscious way. A user who decides to buy an Internet service from an Internet service provider may build her trust offline by asking her friends for recommendations.

Privacy and trust are as closely related in computing environments as they are in social systems [6]. We define *privacy* as an entity's ability to control the availability and exposure of information about itself. This definition extends the scope of privacy from a *person* in the original definition [13] to an *entity*, including an organization or software. The extension is consistent with the use of the notion of "trust" also in relationship to artifacts [1], and with the common practice of antropomorphization of intelligent system components (such as objects and agents) in computer science.

An entity can choose to trade its privacy for a corresponding gain in its partner's trust in it [25]. The scope of a privacy disclosure should be proportional to the expected benefits—a customer applying for a mortgage must reveal much more personal data than one buying a book.

A mere perception of a threat to users' privacy from a collaborator may result in the substantial lowering of trust. This impedes the sharing

Manuscript received January 24, 2005; revised September 14, 2005. This work was supported in part by the National Science Foundation under Grants IIS-0209059 and IIS-0242840, in part by grants from the Defense Advanced Research Projects Agency (DARPA), IBM, and Cisco University Research Program (URP), and in part by a seed Grant from the Regenstrief Center for Healthcare Engineering (RCHE) at Purdue University. This paper was recommended by Guest Editors H. R. Rao and S. J. Upadhyaya.

L. Lilien is with the Department of Computer Science, Western Michigan University, Kalamazoo, MI 49008 USA and affiliated with the Center for Education and Research in Information Assurance and Security (CERIAS) and the Regenstrief Center for Healthcare Engineering (RCHE), Purdue University, West Lafayette, IN 47907 USA (e-mail: llilien@cs.wmich.edu).

B. Bhargava is with the Department of Computer Sciences, CERIAS and RCHE, Purdue University, West Lafayette, IN 47907 USA (e-mail: bb@cs.purdue.edu).

Digital Object Identifier 10.1109/TSMCA.2006.871655

1083-4427/$20.00 © 2006 IEEE