Private and Trusted Collaborations[1]

Bharat Bhargava and Leszek Lilien

Department of Computer Sciences and
Center for Education and Research in Information Assurance and Security (CERIAS)
Purdue University
West Lafayette, IN 47907

{bb, llilien}@cs.purdue.edu

Abstract

Some level of trust must be established before any collaboration or interaction can take place. Since trust and privacy are closely intertwined, a mere possibility of a privacy violation reduces trust among interacting entities. This impedes sharing and dissemination of sensitive data. Affected interactions range from simple transactions to the most complex collaborations. We want to assist users in properly protecting their privacy in such interactions. We also wish to help users give up the minimum degree of privacy necessary to gain the required level of trust—the level demanded by user's partner as a pre-condition for a collaboration. In this paper, we focus on mechanisms for privacy-preserving dissemination of sensitive data. We next consider briefly the issues of privacy metrics and trading privacy for trust. Our test application in the area of location-based routing and services illustrates how to use the proposed privacy-for-trust approaches.

1. Introduction

Lack of trust and violations of privacy impede sharing and dissemination of private data among interacting entities, both humans and smart artifacts. Affected interactions range from simple transactions to the most complex collaborations.

Interactions under considerations involve dissemination of private data, which ranges from voluntary, via "pseudo-voluntary," to mandatory—as required by law. The pseudo-voluntary dissemination is particularly deceitful since it appears to give a user a freedom to decline sharing his private information but, in return, precludes the user from receiving a desirable service. As a simple example, a person who refuses to show his proof of age is not allowed to enter a tavern.

Privacy research is motivated not only by sensitivity of personal data perceived by users. Also, business losses due to privacy violations are growing. Further, many federal and state laws—including the Privacy Act of 1974 and HIPAA of 1996—have been passed to protect privacy.

Trust and privacy are closely intertwined. Even just perceived threats to users' privacy by a collaborator may result in substantial lowering of trust. This could result in rejection of collaboration between prospective partners, and a self-imposed isolation of one or more partners.

For any collaboration—or, even broader, any interaction—a level of trust must be established. The required level of trust varies from low for a simple interaction, to very high for a complex collaboration. Often, especially in simpler cases, use of trust is implicit or transparent. Similarly, quite often trust is used externally, that is outside of the computing environment. For example, a user who decides to buy an Internet service from an ISP builds his trust both externally—by asking his friends for reputable ISPs—and implicitly—by not even considering using trust in a conscious way.

Trust-building measures, expected of a trustworthy partner, include providing quality and integrity of data, and assuring reliable, secure, and privacy-preserving end-to-end data communication. The latter includes sender authentication, guarding message integrity, and using robust network routing algorithms, which can deal with malicious peers, intruders, security attacks, etc.

We want to help users feel that their privacy is properly protected in their interactions, and that they give up the minimum degree of privacy to gain the level of trust demanded as a pre-condition of collaboration by their partners. Appropriate metrics for the assessments of privacy loss and trust gain are a prerequisite for achieving these goals.

This paper proposes solutions for building private and trusted systems and applications. This research can contribute, among others, to cooperative information systems, peer-to-peer collaborations, ad hoc networks, and the Semantic Web. It integrates ideas from privacy, trust, and information theory in database systems as well as communications.

[1] Research supported in part by NSF grants IIS-0209059, IIS-0242840, ANI-0219110, DARPA, IBM, and Cisco URP grant. More information is available at www.cs.purdue.edu/people/bb.

The main results of the presented work are in the following areas: privacy-preserving dissemination of sensitive data, developing privacy metrics, optimizing privacy and trust tradeoffs, and application to trusted routing in wireless networks. They are discussed in turn in Sections 2 through 5 (see also [BhLX04]).

2. Privacy-preserving Data Dissemination

2.1 Problem Statement and Challenges

A *guardian* is either the original owner, or a subsequent stakeholder of sensitive data. A guardian may pass private data to another "lower-level" guardian in a data dissemination chain (actually, this may be a cyclic graph). The risk of privacy violations grows with the chain length and milieu fallibility and hostility.

Traditionally, owner's privacy preferences or policies are *not* transmitted due to neglect or failure. If a privacy policy is not included with data, even an honest receiving guardian is unable to honor them. A conceptually simple solution is encapsulation of policies and other metadata including owner's privacy preferences with owner's sensitive data.

There are two major challenges for this approach. The first is ensuring that owner's metadata are never decoupled from his data. A possible solution involves cryptography, and making data unreadable when its associated metadata is incomplete or absent.

The second major challenge is an efficient protection in a hostile milieu. We need: (a) to consider threats, such as uncontrolled data dissemination, and intentional or accidental data corruption, substitution, or disclosure; (b) effective ways for detecting a loss of data or metadata; and (c) efficient data and metadata recovery methods (for example, a simple recovery by retransmission from the original guardian is trustworthy but inefficient).

2.2 Related Work

The privacy mechanisms for the Web (notably P3P [Cran03]) are not well utilized by service providers. They should be made a part of the data they are supposed to protect not only for Web privacy but also for the entire information technology area [ReBE03]. In this way, they become metadata.

Metadata can be defined as data used for self-descriptiveness [McCK99]. The expressive power of a simple *name-value pair* mechanism for self-describing code/files/programming is demonstrated in [Bent87]. Other examples of the use of self-descriptiveness in different contexts include a metadata model [BoDe03], Knowledge Interchange Format (KIF) language for knowledge bases [GeFi92], components in context-aware mobile infrastructure [Rako99], flexible data types for distributed object systems [SpBe99], and an object-oriented model for meta schema in federated databases [UrAb94].

Data objects passed through the data dissemination chain need protection from malicious guardians. The idea of self-descriptiveness employed for data privacy preservation is briefly mentioned in [ReBE03]. We use it by making the privacy-preservation techniques an integral part of the data it is supposed to protect. Other approaches to protection of a software client (code) from a malicious host include [CoTh00]: (a) *obfuscation*, which protects against reverse engineering of the code, (b) *tamper-proofing*, which protects code against tampering, and (c) *watermarking* the code, including its *fingerprinting*, which protects against software piracy.

An interesting approach to securing mobile self-descriptive objects involves self-destruction. It uses an analogy to one of the two biological cell destruction mechanisms: the chaotic destruction process of *necrosis* due to an injury, and an orderly, programmed destruction process of *apoptosis* [Tsch99]. The latter, in contrast to the former, is "clean"—no toxic substances are leaked to the cell's environment, so no inflammation is induced.

Susceptibility of mobile objects—containing code, a data state, and an execution state—to many types of host attacks is discussed in [SaHS03] and [BGPR97]. Mobile objects or agents can be secured by running on trusted computing platforms. Traditional approach requires a separate, dedicated, tamper resistant platform, for example a secure coprocessor [TyYe94]. In contrast, the Terra virtual machine-based platform [GPCR03] provides the same services on commodity hardware, by partitioning a tamper-resistant hardware platform into multiple, isolated virtual machines, or "closed boxes."

Platform for Privacy Preferences (P3P) is the best-known protocol and a suite of tools for specifying privacy policies of a Web site, and preferences of Web users [Cran03]. P3P is not intended to be a comprehensive privacy "solution" that would address all principles of Fair Information Practices [UFTC98]. AT&T Privacy Bird is a prominent implementation of P3P [APBT04].

2.3 Proposed Approach

We propose a novel comprehensive solution to the *problem of preserving privacy in data dissemination*. Let us consider a simple data dissemination scenario. Suppose that a customer "deposits" his data in a bank. Now the bank immediately encapsulates data within an *atomic* private object, which includes private metadata containing customer's privacy preferences.

With atomic self-descriptive objects there is no way that a sending guardian can transmit to the receiving guardian an object that is incomplete, for example missing some owner's privacy preferences. Whenever delivery of a complete object fails, the receiving guardian can recover it easily by retransmission. This is true for every link of a dissemination chain. This solution solves the problem of preserving privacy in data dissemination for friendly environments, where guardians and their environments are well-behaved, benevolent, and reliable.

The solution must be extended to embrace hostile and unfamiliar environments. In the first step, the extension will involve an atomic *apoptosis*, that is a clean self-destruction, whenever the object feels threatened. A private object is here

a binary-state or atomic entity, which can be either intact or safely destroyed.

In the second step, we generalize the notion of apoptosis with the idea of *object evaporation*. Object's private data are not destroyed all at once but evaporate gradually, in proportion to the object's distrust towards its current milieu. This is an adaptive technique that exploits the environmental context, including the "distance" from the data owner.

We now address the three steps of the proposed solution in turn.

Self-descriptiveness Sensitive data are accompanied within the self-descriptive private objects by their metadata. Comprehensive metadata should include:

- *Owner's privacy preferences*: read and write access circumstances. They include who or what, how, when, etc. is allowed to read or write private data;
- *Guardian privacy policies*: privacy policies of the original and/or subsequent data guardians;
- *Metadata access conditions*: verification and modification circumstances for metadata;
- *Enforcement specifications*: specification enforcement circumstances;
- *Data provenance*: who created, modified, destroyed, or read any portion of data;
- *Context-dependent and other components*: this may include customer trust levels for different contexts, application-dependent elements, and any other elements that are needed for metadata completeness.

The list contents and many elements in the above list are context-dependent. For example, owner's privacy preferences depend on his *trust level* with respect to each guardian. His trust in a less known guardian might be lower than in a better-known one.

For a given object, the policy for a guardian included in *guardian privacy policies* is the one that was in effect at the moment when either the owner or the object "negotiated" its permission to be placed under control of the guardian.

Self-descriptive objects will simplify *notifying* or *requesting permissions* from their owners and guardians, since their contact information is available in the data provenance component. Ideally, owners should always be notified of their data use, and be asked for consent whenever their data are to be accessed in a way that is not predefined in user's preferences or the original guardian's policy [Lang01, Mart01, UFTC98]. The requests and notifications can be sent to owners immediately, periodically, or on demand. Many communication channels are available, including pagers, SMS or, email messages, or conventional mail.

Transmitting *complete* metadata is inefficient. They are extensive since they need to describe all foreseeable aspects of data privacy that can be needed to address privacy issues in any application and under any circumstances. For efficiency reasons, based on the application semantics, only some metadata are carried along. Selected metadata have their scope limited by exploiting application and environment contexts along the data dissemination chain. This issue can be viewed as related to the Semantic Web [BeHL01, Thura03].

Apoptosis (Clean Self-destruction) Use of self-descriptive atomic objects with retransmission recovery solves the problem of preserving privacy in data dissemination for well-behaved, benevolent, and reliable guardians. It is, however, insufficient for addressing the problem for malicious or failure-prone guardian environments. We need to enhance self-descriptive objects with the capability of clean self-destruction. The basic idea is that an object about to be compromised by an attacker or an accident should choose apoptosis over risking a privacy disclosure.

Autonomous *apoptosis mechanism* within an object can be implemented as a set of detectors and triggers—as discussed in [LiBh84, KeSZ02]—setting off associated apoptosis code. The code is activated whenever detectors determine a credible threat of a successful attack on the object.

As with any detectors, false positives and false negatives can occur. Situations in which the self-destruction trigger is overly sensitive can be dealt with by *privacy recovery*. A guardian can recover object from a guardian preceding it. Each recovery results in a performance penalty. The possibility of denial-of-service (DoS) attacks caused by repeated recoveries of the same or different objects has to be dealt with by setting limits on these recoveries.

Objects as a whole must be protected from tampering. The simplest approach is making their embedded detectors more trigger-happy but it increases recovery costs. Other possibilities include using diverse encryption or anti-tampering techniques, including classical code tamper-proofing and obfuscation techniques, and execution of objects only on trusted real or virtual coprocessors.

Proximity-based Evaporation Perfect passing of objects is not always desirable. When data are secretly captured by spyware embedded in browser extensions [Mart01], owners want to see them distorted or destroyed once they leave their computer. Owners are often willing to share their data locally, for example with their colleagues in a lab, but want to avoid their wider dissemination.

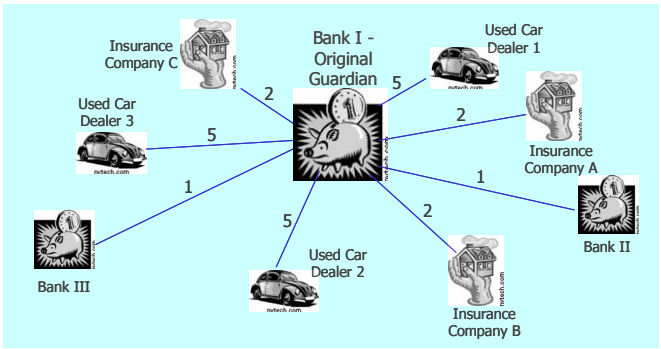

Fig 1. Example of one-dimensional distance: distance proportional to business type.

This suggests that private objects should be *evaporating* in proportion to their "distance" from their source. Owners generally trust their original guardians more than subsequent

ones, further away. Unauthorized data disclosures become more probable at more distant guardians in the chain.

Different context-dependent proximity metrics can be used. For instance, in the business environment people usually feel "closer" to institutions with which they have built trust via a history of satisfying past interactions. A level of trust could be a measure of distance. The "business-type similarity" (cf. Fig.1) could be another metric in situations when customer's trust is related not just to individual institutions but to the type of business. If, for example, a customer trusts banks more than insurance companies, the "distance" from a bank to another bank would be smaller than from a bank to an insurer. Multi-dimensional composite metrics are an option. At least one of the dimensions could be used as a measure of reliability and security of the environment.

Evaporation might be implemented as intentional and controlled object *distortion*. The idea is that appropriately injected noise will make data less valuable or meaningful, and their owners less vulnerable. Examples of data distortion include replacing exact data with approximate data (such as removing the house number from the street address), or up-to-date values with previous values (such as providing not the current account balance but an outdated one).

Evaporation can be seen as a generalization of apoptosis. A complex evaporation implementation, generalizing the apoptosis mechanism, requires making apoptosis detectors, triggers, and code context-dependent in order to enable exploitation of the relevant environmental information. Conventional apoptosis can be implemented as a simple case of data evaporation, in which evaporation follows a step function with a constant maximum value initially, and the zero value above a certain threshold. Such pattern means that the object self-destructs when the proximity metric exceeds a predefined threshold value. This idea can be further extended for designing objects that have the capability to self-destruct when copied onto a "foreign" data storage device or media.

We also investigate the idea of using evaporation within the apoptosis mechanism. As an object drifts further away from its original guardian, its apoptosis trigger can become more sensitive.

3. Privacy Metrics

Problem Statement and Challenges We need privacy metric to determine what degree of data privacy is provided in any combination of users, techniques, and systems.

This gives rise to at least two heterogeneity-related challenges. First, different privacy-preserving techniques or systems claim different degrees of data privacy. These claims are usually verified using ad hoc methods customized for each technique and system. While this approach can indicate the privacy level for each technique or system, it does not allow comparisons of different techniques or systems using various user models.

Second, privacy metrics themselves are usually ad hoc and customized for a user model and for a specific technique or system.

We need a unified and comprehensive privacy measures. A good privacy metric has to compare different techniques/systems confidently. It also has to account for: (a) dynamics of legitimate users—such as how users interact with the system, and awareness that repeated patterns of data access can leak information to a violator; (b) dynamics of violators—such as how much information a violator may gain by watching the system for some time; and (c) costs associated with metric implementation—such as storage, injected traffic, CPU cycles, and delay.

Proposed Approach In [ZhBh04], we propose metrics for assessing the privacy loss. We distinguish the query-dependent and query-independent privacy loss. When evaluating the *query-independent* privacy loss, we assume that a violator is interested in the value of a private attribute. *Query-dependent* privacy loss of a credential nc is defined as the amount of information that nc provides in answering a specific query.

The following examples illustrate the difference between the two types of privacy loss. Let users' age be the private attribute of interest. The first query asks whether a user's age exceeds 15. If we know that a user has a valid driver license, we are assured of the positive answer. The second query asks whether a user is 50 or older—it is a condition to join a silver insurance plan. If we know that a user has a valid driver license, the probability of answering "yes" to the second query is approximately 50% (based on an average age of active driver license holders). This shows that the privacy value of the same piece of information varies for different queries.

We propose two entropy-based probability methods to evaluate the query-dependent and query-independent privacy loss, respectively. The first method evaluates the query-independent privacy loss of disclosing a credential. The second method evaluates query-dependent privacy loss based on the knowledge of the set of potential queries.

More details of both approaches are given in [ZhBh04].

4. Trading Privacy for Trust

Problem Statement and Challenges The increasing adoption of incentive and monitoring mechanisms, including reputation systems, suggests that a highly trusted user can get more benefits, such as discounted prices and better quality of services, from service providers. To quickly gain her partner's trust in an open computing environment—for example on the Internet—a user provides digital credentials, such as certificates, recommendations, or past transaction history. These credentials contain private information, such as a user's identity and shopping preferences. Disclosure of this information reveals some of user's private data.

In real world applications—such as e-commerce and networking applications—users want to build a certain level of trust with the least loss of privacy. This gives rise to the problem: *How to gain a certain level of trust with the least loss of privacy?* A resolution will enable a user to decide

whether to trade some of his privacy for the potential benefits gained from an elevated level of trust.

In more detail, the following research questions need investigation:

- How much privacy is lost by disclosing a specific piece of information? To make things more difficult, information disclosed in the past affects current privacy loss.
- How much does a user benefit by having a higher level of trust? This benefit is referred to as the *trust gain*.
- How much privacy a user is willing to sacrifice for a certain amount of trust gain? User's decision sets the limits on the privacy-for-trust exchange.

There are two main challenges to resolution of the privacy-for-trust problem. First, privacy and trust are fuzzy and multi-faceted concepts, difficult to formalize, quantify, and measure properly. Second, many context-dependent and often conflicting factors affect the amount of privacy lost due to a disclosure. They include: (a) who gets this information, (b) possible uses of this information, and (c) what information was disclosed in the past or is available in the environment.

Proposed Approach We investigated the metrics and algorithms to quantify privacy loss and trust gain. Our assumptions included the following ones: (a) privacy protection satisfies contractual constraints [Clif02], (b) a user has multiple choices on what piece of information to disclose, and (c) each user can make her decision independently.

Our approach included the following steps:

- Formalizing the privacy-trust tradeoff problem
- Estimating privacy loss for a given credential set
- Estimating trust gain for a given credential set
- Devising algorithms minimizing privacy loss for required trust gain

The developed algorithms can be used to either assist a user in her decision making, or fully automate the decision based on policies or preferences predefined by users.

Details of the approach and solutions are presented in [ZhBh04].

5. Preserving Privacy in Wireless Networks with Location-Based Routing and Services

Problem Statement The technological progress that makes positioning devices smaller, cheaper, and more energy-efficient enables a wide deployment of *location-based routing and services* (LBRS) in wireless networks (cf. [HoKa99]. Disclosure of *location* information and *movement patterns* of a mobile node causes users' privacy concerns.

The research problem is to design mechanisms that preserve the location (and movement) privacy of nodes participating in LBRS. A solution must enable nodes to control dissemination of location information. This results in a more secure and scalable routing and improved quality of services in wireless networks, all without sacrificing users' privacy.

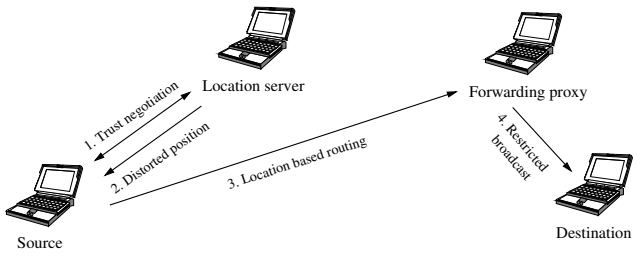

Fig. 2. Basic idea for location-based routing and services.

Proposed Approach The proposed solution (cf. Fig. 2) uses a stale or an offset position as a "shadow" to protect the real physical position. The accuracy of information that a mobile node can get is determined by its trust level. When data packets reach a forwarding proxy in the "shadow" position, the proxy forwards them to their final destinations via restricted broadcast.

Trust Negotiation Between Querying Node and Server A querying node can raise its trust level by showing more credentials to the location server. The node trades its privacy for more accurate location information. The potential benefits include decreases in route length, packet delays, and communication overhead. We can optimize the privacy-for-trust tradeoff to either assist nodes in making their decisions, or make the decisions for them fully automatically.

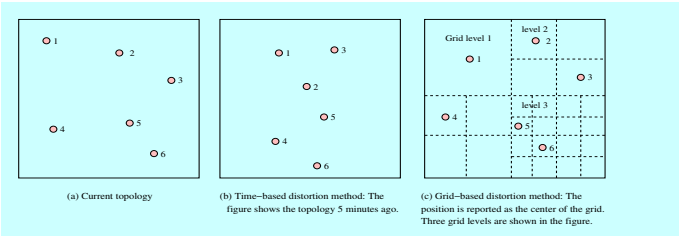

Fig.3. Position distortions for LBRS.

Dynamic Mappings Between Distortion and Trust Level We study two distortion control methods, time-based and grid-based, illustrated in Figures 3a-3b and 3a-3c, respectively. In the *time-based* method, the lower trust level a node has the older location information it receives. In the *grid-based* method, the network area is divided into different-sized grids according to the trust levels. The querying node receives in response not an exact location but only a grid number.

We can define both static and dynamic mappings between trust levels and distortion levels. In the *static* mappings, the movement model and certain node density on the grid may allow the querying node to compromise the privacy. For example, the degree of imprecision is *not* the same for a slow moving node and a fast moving node in case when the positions they had five minutes ago are always returned in response to a position query. In the *dynamic* mappings, a degree of uncertainty for returned location information can be assured to avoid this problem.

We consider the movement model and generate probabilistic queries to guarantee that the destination node cannot be

distinguished within its anonymity set, and thus cannot be traced. The mechanisms for defining the effective anonymity sets are applied in calculation of uncertainty.

More solution details are available in [WaBh04].

6. Conclusions and Future Work

We have presented mechanisms for privacy-preserving dissemination of sensitive data. We also summarized our approaches to defining privacy metrics and trading privacy for trust. Our test application in the area of wireless location-based routing and services illustrates how to use the proposed privacy-for-trust approaches.

We plan to use our testbed system, called PRETTY (Private and Trusted Systems), for extensive experimentation in measurements of privacy, trust and the privacy-trust tradeoffs during various interactions.

Acknowledgements Ms. Yuhui Zhong and Dr. Mohamed Hefeeda contributed to determining privacy metrics, Ms. Yuhui Zhong, again—to research on privacy-trust tradeoff, and Mr. Weichao Wang—to developing the solutions for LBRS.

References

[APBT04] AT&T Privacy Bird Tour: http://privacybird.com/tour/1 2 beta/tour.html. February 2004.

[BeHL01] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, 2001.

[Bent87] J. Bentley. Programming pearls. *CACM*, 30(6):479–483, June 1987.

[BGPR97] F. Bergadano, A. Giallombardo, A. Puliafito, G. Ruffoand, and L. Vita. Security agents for information retrieval in distributed systems. *Parallel Comp.*, 22(13):1719–1731, 1997.

[BhLX04] B. Bhargava, L. Lilien, and D. Xu, "Private and Trusted Interactions," slides, *The 5th Annual Information Security Symposium " Energizing the Enterprise: Cyber Security in Context,"* CERIAS, Purdue University, March 2004; http://www.cs.purdue.edu/homes/llilien/priv_trust_int.pdf

[BoDe03] S. Bowers and L. Delcambre. The uni-level description: A uniform framework for representing information in multiple data models. *ER 2003-Intl. Conf. on Conceptual Modeling, I.-Y. Song, et al. (Eds.)*, pp. 45–58, Chicago, Oct. 2003.

[Clif02] C. W. Clifton. Tutorial on privacy, security, and data mining, 13th European Conf. on Machine Learning (ECML'02) and 6th European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD'02), 2002.

[CoTh00] C. Collberg and C. Thomborson. Watermarking, tamper-proofing, and obfuscation—tools for software protection. Tech. Rep. 2000-03, Dept. of Comp. Sci., U. of Arizona, 2000.

[Cran03] L. Cranor. P3P: Making privacy policies more useful. *IEEE Security and Privacy*, pp. 50–55, Nov./Dec. 2003.

[GeFi92] M. Gensereth and R. Fikes. Knowledge interchange format. Tech. Rep. Logic-92-1, Stanford Univ., 1992.

[GPCR03] T. Garfinkel, B. Pfaff, J. Chow, M. Rosenblum, and D. Boneh. Terra: A virtual machine-based platform for trusted computing. *ACM Symp. on Oper. Syst. Principles (SOSP)*, Oct. 2003.

[HoKa99] T. Hodes and R. Katz. Composable ad hoc location based services for heterogeneous mobile clients. *Wireless Networks*, 5(5):411–427, 1999.

[KeSZ02] F. Kerschbaum, E. H. Spafford, and D. Zamboni. Using internal sensors and embedded detectors for intrusion detection. *J. of Computer Security*, 10(1/2):23–70, 2002.

[Lang01] M. Langheinrich. Privacy by design - principles for privacy-aware ubiquitous systems. *UbiComp*, Atlanta, GA, Oct. 2001.

[LiBh84] L. Lilien and B. Bhargava. A scheme for batch verification of integrity assertions in a database system. *IEEE Trans. on Software Engineering*, SE-10(6):664–680, Nov. 1984.

[Mart01] D.M. Martin Jr., et al. The privacy practices of web browser extensions. *CACM*, 44(2):45–50, Feb. 2001.

[McCK99] R. McClatchey, Z. Kovacs, F. Estrella, J.-M. L. Goff, L. Varga, and M. Zsenei. The role of meta-objects and self-description in an engineering data warehouse. *Proc. Intl. Database Engineering and Applications Symposium (IDEAS 1999)*, pp. 342–350, Montreal, Canada, Aug. 1999.

[Rako99] A. Rakotonirainy. Trends and future of mobile computing. *10th Intl. Workshop on Database and Expert Systems Applications*, Florence, Italy, Sept. 1999.

[ReBE03] A. Rezgui, A. Bouguettaya, and M. Eltoweissy. Privacy on the Web: Facts, challenges, and solutions. *IEEE Security and Privacy*, pp. 40–49, Nov./Dec. 2003.

[SaHS03] M. Saeb, M. Hamza, and A. Soliman. Protecting mobile agents against malicious host attacks using threat diagnostic and/or tree. *Proc. Smart Objects Conf.*, Grenoble, France, May 2003.

[SpBe99] M. Spreitzer and A. Begel. More flexible data types. *Proc. IEEE 8th Workshop on Enabling Technologies (WETICE '99)*, pp. 319–324, Stanford, CA, June 1999.

[Thura03] B. M. Thuraisingham. Security issues for the semantic web. *COMPSAC*, pp. 632–637, 2003.

[Tsch99] C. Tschudin. Apoptosis - the programmed death of distributed services. In: J. Vitek and C. Jensen, eds., *Secure Internet Programming*. Springer-Verlag, 1999.

[TyYe94] J. D. Tygar and B. Yee. Dyad: A system for using physically secure coprocessors. *Proc. IP Workshop*, 1994.

[UFTC98] U.S. Federal Trade Commission. Privacy online: A report to congress. June 1998.

[UrAb94] S. Urban and T. B. Abdellatif. An object-oriented query language interface to relational databases in a multidatabase database environment. *Proc. Intl. Conf. on Distributed Computing Syst. (ICDCS 1994)*, pp.387–394, Poznan, Poland, June 1994.

[WaBh04] W. Wang and B. Bhargava, "Location Privacy in Geographical Routing for Mobile Ad Hoc Networks," (poster paper), *Proc. SKM 2004*, Amherst, NY, Sept. 2004.

[ZhBh04] Y. Zhong and B. Bhargava, "Using Entropy to Trade Privacy for Trust," *Proc. SKM 2004*, Amherst, NY, Sept. 2004.