Can Ferris Bueller Still Have His Day Off? Protecting Privacy in the Wireless Era

Ben Greenstein[*] Ramakrishna Gummadi[†] Jeffrey Pang[§]
Mike Y. Chen[*] Tadayoshi Kohno[‡] Srinivasan Seshan[§] David Wetherall[‡*]
[*]Intel Research Seattle [†]University of Southern California
[‡]University of Washington [§]Carnegie Mellon University

Abstract

Today's rich and varied wireless environment, including mobile phones, Wi-Fi-enabled laptops, and Bluetooth headsets, poses threats to our privacy that cannot be addressed with existing protocols. By considering 802.11 as a case study and analyzing publicly available 802.11 traces, we show that a device can be identified and tracked over time through its persistent link-layer address, list of known networks (SSIDs), and other protocol and physical layer characteristics. We argue that it is in the best interest of providers as well as users to design systems that maintain user privacy. We identify several research challenges to doing so and offer some direction towards a solution.

1 Introduction

Many people now have several wireless devices, and new ones—like the Wii controller and the Nike+iPod pedometers—are regularly being introduced and adopted. Perhaps the biggest reasons for their success are that they support mobility and are convenient. With more and more locations providing wireless services (*e.g.*, metropolitan-scale 802.11 networks are deployed in Taipei and are under way in San Francisco, Moscow, and many other cities [21]), users can be mobile while staying connected.

This heightened level of wireless connectivity brings many advantages, yet it also threatens our privacy in new ways that are underappreciated. It is well-known that wireless links are more exposed than their wired counterparts, as messages are broadcast to anyone within radio range (250 meters on 802.11b with standard antennas, and much farther to receivers with directional ones [19]). To counter the threat of eavesdropping and provide confidentiality comparable to what one might expect on a wired network, privacy mechanisms such as WEP/WPA for 802.11 and A5/1 for GSM encrypt packet contents. Although such mechanisms have been plagued with design and implementation flaws [6, 7], we take the optimistic view that the problem of how to build secure links is solved from the point of view of research and we do not consider it further in this paper.

The new threats of wireless devices are highlighted by recent concerns about *location privacy*. This risk stems from the mobility of wireless devices. The fear is that people can be tracked every second of every day, and with high accuracy, via the devices they carry. Location-aware systems, such as Active Badge [25] and RADAR [3], which explicitly estimate and communicate device positions, have spurred significant research in the mobile and ubiquitous computing communities focused on preventing the unauthorized disclosure of this information. Solutions include using centralized access controls (*e.g.*, [13, 14, 22]), having sensors perturb collected data before it is stored [12], and enabling users to discover their own locations using only passive measurements [18].

Location privacy is also threatened by systems that do not provide location awareness explicitly; any wireless device may betray who and where a user is, because transmissions often contain unique identifiers (*e.g.*, RFID tags [16]) or addresses (*e.g.*, 802.11 and Bluetooth devices [11, 15, 28]) that can be observed by anyone nearby. Moreover, masking these identifiers using *pseudonyms* [11, 15] or *mix zones* [5] is problematic in practice when they are also used for authentication and/or billing, as MAC addresses are in some 802.11 networks.

We argue that location privacy threats should all be viewed as facets of a larger wireless privacy problem; the RFID, 802.11 and Bluetooth tracking threats are essentially equivalent in their reliance on eavesdropping to discover unique identifiers or addresses. This threat should be recognized whether locations are made explicit as part of the operation of the system or can be unintentionally inferred. And it is posed (to different degrees) by service providers and legitimate users as well as third-party eavesdroppers. Note that current best security practices for data confidentiality do not alleviate these threats.

Moreover, wireless privacy threats are broader than the tracking of specific individuals. This is because even more limited information leaks can be of concern. As wireless devices become more diverse and more specialized in their individual function, it becomes easier to infer what kind of devices or applications are in use at a particular location, irrespective of the identity of the user. This threat (known as inventorying in the context of RFID) may be used to profile people, *e.g.*, for health via wireless medical or fitness devices, or target individuals, *e.g.*, for theft of Zunes or expensive home entertainment systems. Even the detection of many forms of wireless communication leaks valuable information about the nearby environment due to small transmission ranges. Monitoring may reveal whether it is likely that people are present; this may violate users' natural assumptions about the privacy of their environments, *e.g.,* when their presence in their homes is detected from outside. Monitoring also may be used to estimate population or activity levels [23].

We argue that any systematic treatment of wireless privacy must encompass all of the above threats rather than view them piecemeal. Efforts to do so will face several challenges. They must articulate the problem in a way that goes beyond isolated examples of privacy failures; we hope this position paper contributes to this step. Then they must devise solutions that protect privacy and cause them to be put into practice. We make two points relevant to these steps. The first is that technical solutions are likely to play a large role in solutions in addition to legal, regulatory and social mechanisms. Wireless privacy threats are a technical creation, and we use a case study of 802.11 in this paper to argue that technical changes in the design of wireless protocols can offer some immediate relief. This is beneficial because it sidesteps the costs of other solutions, and because legal deterrents are less likely to be viable for many forms of wireless communications, *e.g.,* in the unlicensed ISM band versus cellular systems.

The second point is that, while privacy is often portrayed as a user concern that is unevenly appreciated, it is in the interests of providers to deploy systems that respect privacy. This is because there are legal and financial risks associated with the disclosure of confidential customer or employee information, whether by accident, theft, or subpoena. This is apparent from the public outcry surrounding privacy invasions, *e.g.,* the AOL release of Web search terms that identified users and the Benetton RFID boycott, and regulatory requirements on some businesses such as telecommunications providers. Moreover, companies are increasingly developing policies to protect the private information of their employees in all facets of operations as society is increasingly aware of privacy threats.

In the remainder of this paper, we highlight possible wireless privacy threats using an analysis of publicly available 802.11 traces, describe measures that can begin to improve privacy, and discuss some of the longer-term technical challenges that must be addressed. We choose 802.11 for our case study because our results, and in particular a new privacy attack that we uncover, suggest that the privacy problem for wireless networks may be more complex than originally anticipated.

2 802.11 Case Study

To make our discussion concrete, we sketch several scenarios that highlight existing privacy threats in the context of 802.11 networks. Each scenario illustrates a different type of threat and exposes a different privacy leakage vector. These fictional scenarios involve FooNet, a metropolitan-scale network provided by Foo, and two employees of major companies: Ferris from Foo and Boris from Bar. The threats range from individuals (both users of FooNet and other parties in the vicinity) to providers (Foo and FooNet in this case) as both victim and attacker.

To back up these scenarios, we studied 802.11 traces taken at SIGCOMM in 2004 to understand what information is leaked. These traces have been anonymized to protect the identities of the attendees: client MAC addresses were consistently hashed and the contents of data frames were removed. In our exercise, we let the hashed MAC addresses identify individual clients, and assume that the payload is not available to any parties other than the client and access point (AP). The latter would be the case if encryption keys were established per client as in WPA2, but admittedly this was not used at SIGCOMM 2004.

Scenario 1: Provider threat to individual privacy. Ferris bought a ZuNod, a Wi-Fi-enabled portable music player, during lunch on Monday, and spent the afternoon at work setting it up. By the end of the day he was able to connect the ZuNod to Foo's corporate Wi-Fi network, authenticate using his corporate username and password, and download songs from his favorite site.[1] On Tuesday morning, Ferris calls in sick to spend the day listening to music while strolling around the city. He subscribes to FooNet's free Internet service to download songs while mobile.

For amusement, an overeager Foo human resources associate decides to search for MAC addresses that connect to both Foo's corporate network and FooNet. The result isn't particularly interesting. A lot of employees use FooNet. However, when he restricts the results to the MAC addresses used by employees who have called in sick, he finds Ferris. Moreover, on closer inspection, he sees that Ferris spent much of the day in the city park and at the art museum. Nothing is done with this information as it is clearly an invasion of Ferris' privacy. However, by coincidence, Ferris is fired two weeks later. Ferris' lawyers subpoena HR records, find the MAC address search, sue Foo for invasion of privacy, and a media storm ensues.

Problem 1: Persistent addressing. The above scenario is possible because 802.11 interfaces broadcast persistent and globally unique MAC addresses, which is a known privacy problem. This persistence makes them easy to use to deliver packets, and their uniqueness helps avoid problems associated with naming collisions. However, such addresses also allow separate observations of the same client at different places and times to be tied to-

[1] The ZuNod is a hypothetical Zune-like device with unfettered Wi-Fi access. The initial Zune restricts Wi-Fi usage to local music sharing.

gether. This allows not only tracking by any observer [15] but linkage of other databases. This is what enables Ferris to be profiled even though he maintains corporate and FooNet accounts that are otherwise unrelated. Clearly, if we have any hope of providing wireless privacy, we must obscure these addresses.

Scenario 2: Individual threat to individual privacy. Boris' job in the venture capital group at Bar is to meet with startups and decide which ones to fund. He'll usually arrange several onsite visits to those startups that have compelling business plans. When away from his office, Boris opportunistically connects to available open networks and uses a VPN to encrypt his data.

Before being fired, Ferris had the same job at Foo and had an uncanny knack for beating Boris to the good startup opportunities. Ferris and Boris often ate lunch at a restaurant that provides free Wi-Fi access. While Boris would spend his lunches catching up on e-mail and reading the news, Ferris would spend them monitoring to see to which networks Boris's computer tries to connect. This would tip Ferris off to likely startup locations, which he would later visit to look for funding candidates.

Problem 2: Exposed resource discovery. We discovered that the above scenario is possible through trace analysis. Many 802.11 clients actively scan for specific networks to which they have connected in the past. They do this by sending probe request frames, each of which contains the SSID name of one of the networks that it prefers to use. These SSIDs are sent in the clear. This probing behavior is the default for Windows XP as it speeds up network discovery and provides a way to associate with networks that don't periodically announce their existence. Hence Ferris can observe Boris' preferred SSIDs by listening to his transmissions; any observer can do this despite data encryption. Ferris can then obtain the likely street addresses by looking the SSIDs up in a Wi-Fi location database such as Wigle [27].

This scenario demonstrates that while hiding addresses is necessary to provide wireless privacy, it isn't sufficient because other kinds of names are exposed. For active scanning in 802.11, the network names are often revealing, even without a database such as Wigle, since providers often choose meaningful names. Thus when a user probes for the networks that he has connected to previously, in effect, he advertises where he *has been*; *i.e.*, an attacker could use this technique to compromise a victim's *past* privacy.[2] Moreover, an unusual SSID or set of SSIDs can alert an observer to the presence of a particular user, regardless of whether the user's MAC address is changed frequently. Active scanning was previously known to be a security flaw as it can facilitate hijackings [20], but it had not been associated with privacy leaks as best we know. In the traces we examined, 161 users emitted a network name that was unique to a single user. 460 of the 566 users in the trace advertised the names of their preferred networks.

Scenario 3: Provider/individual threat to provider privacy. Bar plans to compete with FooNet by launching its own metropolitan scale network. In an effort to take advantage of Foo's successes and mistakes, Bar decides to analyze FooNet to learn how many users connect to it, how much traffic they generate, and so forth. It is in Foo's best interest to keep these statistics to itself to prevent Bar from gaining a second mover advantage. To monitor FooNet, Bar deploys a much smaller number of mobile nodes that drive by FooNet hotspots at various times of day. Similarly, Boris might drive around himself and provide this information to one of his startup ventures. A comparable threat may be looming in the city-wide 802.11 joint venture in San Francisco [1], which calls for two service providers with different business models (fee-based versus advertising-driven) to share some physical infrastructure, including APs and backhauls. Pricing and service offerings of one company might be determined in part by access statistics that are gleaned from the other.

Problem 3: Apparent network usage. This scenario is possible because various 802.11 packets and fields leak information about network usage. For example, a sequence number field is typically incremented by the sender for each packet that is transmitted. This enables a small number of packets from the AP to be used to gauge the rate of packet transmissions. In beacon frames, which are typically sent by the AP ten times per second, there is a traffic indication map for clients using power-save functionality (which is becoming increasingly popular as a means of extending battery life on small devices). This allows the number of power-save clients to be readily counted, the size of the overall client pool to be estimated from statistics on power-save usage, and perhaps even the length of client sessions to be gauged. These techniques allow Bar to estimate AP usage by observing a relatively small number of packets; any observer can do likewise since these fields are not encrypted. Of course, sampling will always allow a subset of packets to be observed to estimate the whole. Our point is that the current design of 802.11 makes sampling strategies especially effective because 802.11 packets leak much more network usage information than is necessary.

3 Research Challenges

Wireless privacy means more than concealing the contents of a wireless communication, but a precise definition and good metrics that the community can agree upon are elusive because the situation is complicated. Our notion of privacy changes according to who might

[2] We actually found this vulnerability when we noticed that the SSID of one of our own home networks was evident in the trace! Looking further, we were surprised to find that the trace named other networks at universities and companies that one of us had visited before attending SIGCOMM 2004.

be listening. A user might willingly reveal his identity to his provider, but not to other users. And the targets of privacy attacks may range from individuals and their devices to whole companies. Moreover, though several methods have been proposed to measure privacy, such as anonymity sets [5], entropy [9], and *k*-anonymity [24], there's no agreed-upon threshold for being private enough.

The scenarios presented in the previous section underscore the importance of protecting privacy and highlight several challenges: Ferris was uniquely identified through his ZuNod's MAC address, Boris's clients were revealed in SSIDs, and Bar studied FooNet's operations by monitoring sequence numbers and other information sources. This section discusses three technical challenges immediately relevant to these scenarios: First, how can we balance the need for names to address devices with the desire to prevent names from being identifying? Second, how can we discover and bind to resources without revealing that we are doing so or have done so in the past? Third, might even the physical characteristics of transmissions and the control information contained within them leak other subtle and implicit identifying information, and if so, do we have any hope of designing media access protocols that preserve privacy completely? While we phrase this discussion in the context of our 802.11 case study, we believe that these challenges are broadly applicable to other wireless protocols as well, such as Bluetooth, Zigbee, and WiMAX.

Naming. Network addresses identify communicating endpoints. If they are persistent, they can be used to link multiple packets to the same user. 802.11 uses unique MAC addresses that do not change over time and that are broadcast in the clear. Several potential approaches reduce the threat protocols like 802.11 pose to a user's anonymity, but all increase complexity and computation overhead.

Periodically changing MAC addresses, effectively creating temporary pseudonyms as proposed by Hu and Gruteser [11, 15], would increase the difficulty for both users and service providers to link packet transmissions to a source. In 802.11, this would require only user-level changes to the client, so long as the period between changes is large. For example, a user-level script could change a client's MAC address before each AP association. This approach could be extended to change addresses more often and while associated. The client could generate a new MAC address, re-associate with the AP under that address, and send a gratuitous ARP to establish a binding between his IP address and the new MAC address.

A client might want to use this technique to generate a new pseudonym for each frame it transmits to improve anonymity, but this might be infeasible without first making significant changes to the network stack, as throughput would decrease due to temporarily undeliverable packets and additional messaging. Luckily, many common types of wireless traffic, HTTP for example, are short-lived and spaced out over time, so a user could improve his experience by rolling MAC addresses only when his device is idle; of course, the network he uses will still lose bandwidth to the ARP messages he sends.

However, the risk when using pseudonyms is that they can be linked together, and this is achieved easily when some information carried on a client's transmissions remains constant while his pseudonyms are changing. For example, a client's IP address can be used to link the previous MAC address to a current one. A confidentiality scheme such as WPA2, which encrypts all link-layer data payloads, including ARP messages, can be used, but even this would hide only MAC-to-IP bindings from adversaries who lack network privileges; any eavesdropper that is associated with the wireless network can see gratuitous ARP responses and can send ARP requests. As another example, many machines (*e.g.,* any machine with iTunes music sharing enabled) now enable multicast DNS and DNS service discovery; thus eavesdroppers can now use local-link DNS requests to discover machine name to IP bindings [8]. An effective pseudonym scheme would require coordination across network layers, such as by synchronizing name changes. Note that a consistent identifier is not strictly necessary to link pseudonyms as the sudden cessation of one address and use of another may be suggestive, especially when bolstered by other physical layer characteristics such as signal strength. We explore this non-naming related tracking later.

As well as considering naming and information leakage *vertically* up the network stack, effective privacy solutions must consider leakage *horizontally* across network interfaces and devices. Since many user devices now have multiple radios, such as 802.11 and Bluetooth, an eavesdropper can leverage the persistence of any name on one interface to link the changing names of another. This problem, of course, extends to the multiple radio interfaces of the multiple devices a user might be using simultaneously.

A better approach to hiding persistent identifiers might be to encrypt the addresses, perhaps with a nonce so that successive encrypted identifiers would not be identical. If public or pair-wise shared keys are used, addresses could be hidden even from other authorized users of the network. Unlike the pseudonym approach, however, encryption alone would not prevent a client's communicating peer from learning its identity. Other cryptographic approaches would be needed to provide this added privacy. It would be challenging to adapt cryptographic schemes to the task of address obfuscation, without requiring significant changes to existing media access protocols and without incurring excessive overhead.

Discovering resources and binding. Our wireless devices rely on the ability to discover and bind to services on the fly. We consider several privacy goals relevant to this process, which are challenging to meet. First, only clients who are authorized to use a private service should be capable of learning of its presence. The presumption is that an authorized client would know *a priori* of its existence. Second, at most the client and the service involved should know when a binding is established or broken between them; optionally, the identity of the client may be hidden from the service as well. This would prevent adversaries, such as competing providers, from learning information about how and when a service is used. It is evident from Section 2 that today's 802.11 implementations reveal this information during discovery and binding as they leak SSIDs. Obscuring SSIDs in a simple way, such as by hashing them, might render them unreadable, but they would still be consistent, and could therefore be profiled or mapped offline. Third, a solution that provides private resource discovery and binding should be secure from common attacks, such as man-in-the-middle, spoofing, and replay, as well as be compatible with existing media access protocols.

To achieve these goals in 802.11, one might design an anonymous messaging scheme in which the contents of all management frames used for service discovery, authentication, and binding (*i.e.*, association) are encrypted with either public or shared keys; moreover, packet lengths and remaining cleartext fields, such as the frame types, could be made homogenous. However, a comprehensive design additionally would need to deal with several serious challenges that arise in practice. These include: bootstrapping cryptographic state at autonomous clients; disseminating and consistently managing such state at APs; ensuring system scalability as the number of clients increases; synchronizing cryptographic state between a client and an AP in the face of message loss caused by wireless links; ensuring that the encryption operations do not leak the identities of the intended recipients (since standard public key encryption schemes need not provide key-privacy [4]); and making sure that the resulting scheme has acceptable overheads.

As a strawman approach, consider a client who knows the public keys or identities of all the APs to which it may wish to associate. The client could send an encrypted probe to each AP using an *anonymous* public key [4] or *anonymous* identity based [2] encryption scheme. The probe could include a (possibly ephemeral) public key for the client, and the target AP could use this public key to encrypt the response. This strawman approach is computationally heavy for both the client and the APs; the anonymity property of the encryption schemes means that the APs must perform non-trivial computations on each encrypted probe, even if the probe is intended for a different AP. Our strawman approach shares commonality with the randomized hash lock protocol for anonymous authorization [26] in which an RFID tag reader must try all tag keys in order to determine the identity of an RFID tag. To improve upon this strawman, one might consider amortizing the cost of expensive cryptographic operations over multiple sessions, albeit with a potential degradation in the level of privacy provided. For example, after binding with a resource once, a client might maintain a single use or time-limited token that would efficiently catalyze the client's next attempt to discover that resource. As in [15], the token itself may be blinded to prevent the AP from linking two clients together over time.

Limiting information leakage. Explicit names aren't the only identifiers. Other information conveyed in a wireless link layer protocol might reveal clues to a user's identity to varying degrees. The challenge is to find such *implicit identifiers*, measure their utility, and when necessary, devise strategies to remove them while retaining useful functionality and without noticeably degrading performance.

802.11 frames reveal sequence numbers, operating modes, and capabilities to anyone listening, which can be used to link multiple packets to the same source as well as to narrow down the range of the transmitter's possible identities. This information is found in frame headers and control and management frame payloads, but even the best confidentiality schemes cover only the payloads of data frames. An obvious privacy solution would be to encrypt the entirety of every type of frame, but doing so might not be as easy as it seems. First, some link header fields are designed to be broadcast to all users. For example, the duration field is used to announce to all contending users how long the channel will be in use. Encrypting such a field might require using a key that is shared by all authorized users, and would thus be defenseless against an attack by an authorized user; FooNet and other metropolitan-scale networks have thousands of authorized users, thus sharing a key among them wouldn't provide much protection. Second, if a client were to encrypt the remaining fields so that only the AP could decrypt them, then the AP would suffer additional computation load, and would thus be more susceptible to denial-of-service attacks.

Encryption, moreover, might be insufficient. Wireless protocols are often defined to support a number of configuration parameters, be extensible, and provide slack. This gives device manufacturers, driver developers, and end-users a wide range of usage options. In general, this configurability encourages innovation, reduces costs, and improves performance, as communication can be tailored to a particular environment or application in use. However, when a device uses a protocol in a slightly different way—exercising some options and not others—it makes it easier to profile and fingerprint it. For example, we easily can differentiate a device that uses power-save mode or virtual carrier sensing (RTS/CTS) from one that never does. Likewise, it is easy to tell apart two devices that send background probe requests every 60 and 60.2 seconds respectively.

Identifying information may also be found in protocol timings, system and network card clock skew, radio frequency fingerprints, how the client chooses a data transmission rate in response to prevailing conditions, data transmission timings (e.g., the periodicity of probe requests), how the received signal strengths of multiple packets can be linked to the same sender, and how packet lengths are distributed. This information may be found in single packet transmissions, as well as by profiling a group of packets that are presumed to be from the same source. Note for the latter, to ensure privacy, it might be sufficient only to obscure the linkings of packets (to prevent grouping them in the first place). Although some work has demonstrated that these leaks may be used to identify which device driver is in use [10] or otherwise fingerprint a device [17], it remains to be seen whether they provide enough identifying information to profile users uniquely. If they do, then the challenge is not only to plug them, but to do so efficiently. The remedies against timing attacks, packet length profiles, and signal strength correlation might be to induce delays, pad packets, and adjust transmission powers; unfortunately, these solutions would all result in reduced throughput.

4 Conclusions

This paper argues that wireless networks pose new threats to privacy and that there is incentive, not just for users, but also for providers and manufacturers to address these threats. In an effort to explain how to make these networks private, we explain how one might identify the presence of particular users or types of devices, both by looking at explicit identifiers such as names and addresses which are used to discover networks and deliver data, and by profiling users' communications to develop implicit identifiers. Furthermore, we argue that prior efforts to anonymize communication still leak explicit identifiable information during service discovery. It seems that many research challenges remain to be solved before Ferris can have his day off!

References

[1] EarthLink and Google win San Francisco Wi-Fi bid. `http://news.com.com/EarthLink+and+Google+win+San+Francisco+Wi-Fi+bid/2100-7351_3-6058432.html`.

[2] M. Abdalla, M. Bellare et al. Searchable encryption revisited: Consistency properties, relation to anonymous IBE, and extensions. In *Crypto'05*, 2005.

[3] P. Bahl and V. N. Padmanabhan. Radar: An in-building RF-based user location and tracking system. In *INFOCOM*, 2000.

[4] M. Bellare, A. Boldyreva, A. Desai, and D. Pointcheval. Key-privacy in public-key encryption. In *ASIACRYPT'01*, pages 566–582. Springer-Verlag, 2001.

[5] A. R. Beresford and F. Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 2(1):46–55, 2003.

[6] A. Biryukov, A. Shamir, and D. Wagner. Real time cryptanalysis of A5/1 on a PC. In *FSE '00: Proc. 7th International Workshop on Fast Software Encryption*, pages 1–18. Springer-Verlag, 2001.

[7] N. Borisov, I. Goldberg, and D. Wagner. Intercepting mobile communications: The insecurity of 802.11. In *Proc. MobiCom*, pages 180–189, 2001.

[8] S. Cheshire and M. Krochmal. Multicast DNS, Aug 2006. URL `http://www.multicastdns.org/`. IETF Draft.

[9] C. Diaz, S. Seys, J. Claessens, and B. Preneel. Towards measuring anonymity. In *Workshop on Privacy Enhancing Technologies*, volume 2482 of *LNCS*, 2002.

[10] J. Franklin, D. McCoy et al. Passive data link layer 802.11 wireless device driver fingerprinting. In *Proc. 15th USENIX Security Symposium*, pages 167–178, jul-aug 2006.

[11] M. Gruteser and D. Grunwald. Enhancing location privacy in wireless LAN through disposable interface identifiers: A quantitative analysis. *Mob. Netw. Appl.*, 10(3):315–325, 2005.

[12] M. Gruteser, G. Schelle, A. Jain, R. Han, and D. Grunwald. Privacy-aware location sensor networks. In *HotOS IX*, 2003.

[13] U. Hengartner and P. Steenkiste. Access control to information in pervasive computing environments. In *HotOS IX*, 2003.

[14] U. Hengartner and P. Steenkiste. Access control to people location information. *ACM Trans. Inf. Syst. Secur.*, 8(4), 2005.

[15] Y.-C. Hu and H. J. Wang. A framework for location privacy in wireless networks. In *Proc. ACM SIGCOMM Asia Workshop*, April 2005.

[16] A. Juels. RFID security and privacy: A research survey. *IEEE Journal on Selected Areas in Communication*, 24(2), Feb. 2006.

[17] T. Kohno, A. Broido, and k. claffy. Remote physical device fingerprinting. In *IEEE Symposium on Security and Privacy*, pages 211–225, May 2005.

[18] A. LaMarca, Y. Chawathe et al. Place Lab: Device positioning using radio beacons in the wild. In *Proceedings of Pervasive 2005*, 2005.

[19] J. Li, C. Blake, D. S. D. Couto, H. I. Lee, and R. Morris. Capacity of ad hoc wireless networks. In *MobiCom '01: Proc. 7th annual international conference on Mobile computing and networking*, pages 61–69. ACM Press, 2001.

[20] Microsoft. Wireless client update for Windows XP with service pack 2. URL `http://support.microsoft.com/kb/917021`.

[21] Muniwireless. URL `http://www.muniwireless.com/`. MuniWireless is the portal for news and information about citywide wireless broadband projects around the world.

[22] G. Myles, A. Friday, and N. Davies. Preserving privacy in environments with location-based applications. *IEEE Pervasive Computing*, 2(1), 2003.

[23] E. O'Neill, V. Kostakos et al. Instrumenting the city: Developing methods for observing and understanding the digital cityscape. In *Ubicomp*, pages 315–332, 2006.

[24] L. Sweeney. k-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, 2002.

[25] R. Want, A. Hopper, V. F. ao, and J. Gibbons. The active badge location system. *ACM Trans. Inf. Syst.*, 10(1), 1992.

[26] S. A. Weis, S. E. Sarma, R. L. Rivest, and D. W. Engels. Security and privacy aspects of low-cost radio frequency identification systems. In *Security in Pervasive Computing*, volume 2802 of *Lecture Notes in Computer Science*, pages 201–212, 2004.

[27] Wigle. URL `http://www.wigle.net/`. Wireless Geographic Logging Engine: Making maps of wireless networks since 2001.

[28] F.-L. Wong and F. Stajano. Location privacy in Bluetooth. In *ESAS*, pages 176–188, 2005.