Tackling Security Vulnerabilities in VPN-based Wireless Deployments

Lookman Fazal[*], *Sachin Ganu*[#,1], *Martin Kappes*[*], *A. S. Krishnakumar*[*], *P. Krishnan*[*]

[*]Avaya Labs Research
233 Mt. Airy Road, Basking Ridge, NJ 07920, USA

[#]WINLAB, Rutgers University
73 Brett Rd, Piscataway, NJ 08854, USA

Email: fazall@avaya.com, sachin@winlab.rutgers.edu,{mkappes, ask, pk}@avaya.com

***Abstract*— Current "best practice" recommendations for enterprise wireless deployments suggest the use of VPNs from a wireless client for both authentication and privacy. In this paper, we demonstrate a security issue with such deployments, which we refer to as the *hidden wireless router* vulnerability. This vulnerability is inherent in the VPN-based wireless LAN architecture, and leads to unsuspecting clients becoming conduits for an attack, exploiting features readily available in popular operating systems like Windows™ and Linux. We describe the attack scenario, and possible solutions for both detecting and locating such hidden wireless routers. Our solutions include a range of possibilities stretching from purely passive to active probing methods, and Access Point-based solutions. We describe our techniques and results of our implementation and experiments.**

Keywords: Wireless LANs, Security, VPNs, Hidden wireless router, Vulnerability.

I. INTRODUCTION

With the proliferation of 802.11- cards, and laptops with in-built 802.11- chipsets [1], the demand and feasibility of universal access to wireless networks is a reality. Adequate security has been a major issue in the deployment of enterprise wireless networks. It has been long understood that direct wireless access to the corporate intranet defeats the deployment of security tools like firewalls and intrusion detection systems (IDSs) that are designed to protect the enterprise network from the Internet [2]. Wireless is also an easily accessed "open" medium, inviting attacks from anybody physically close to (i.e., within radio range of) the enterprise location.

Initial deployments of 802.11 networks used wired equivalent privacy (WEP) to secure communication. It is well understood now that WEP has serious drawbacks and is inadequate for security [3][4][5][6]. In response to the need for security in wireless networks, the IEEE 802.11 working group instituted Task Group *i* to produce a security upgrade for the 802.11 standard. This new standard, namely 802.11i [7], is based on 802.1X port-based authentication [8] for users and devices, and addresses most of the issues with WEP. In particular, 802.11i provides per-user authentication, per-session (cryptographically) strong key(s), and other desirable features. 802.11i has undergone some revisions to incorporate fixes to known security problems [9]. It is expected that 802.11i-based devices will reach the marketplace soon. However, it will take a while before 802.11i is widely deployed.

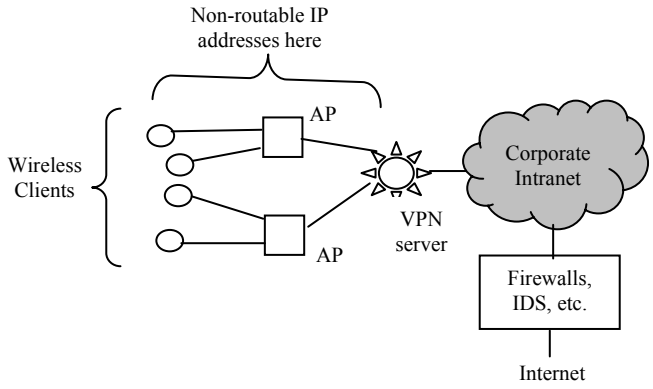

Figure 1. Current Wireless Deployment Strategy

In the meantime, many enterprises are using a VPN-based architecture as the "best practice" method to secure their wireless networks [10][11]. As shown in Figure 1, the wireless and wired networks are separated using a VPN server. Clients are configured to use WEP to associate with access points (APs); however, it is assumed that WEP does not provide any specific level of security. Upon association, the client obtains a *non-routable* IP address (e.g., 192.168.1.32) using DHCP. The client then initiates a VPN connection to the VPN server (e.g., 192.168.1.1). The VPN client on the user's laptop is usually the same client that the user employs to access the corporate network remotely (e.g., from home). After appropriate authentication and key exchanges, a secure tunnel is established for communication and access to the corporate Intranet. Establishing the VPN requires per-user authentication. Clearly, all benefits of VPNs (security, privacy, etc.) are obtained with this setup. In particular, the packets on the air are encrypted, and provide as good privacy as IPSec does.

The VPN-based architecture is motivated by its simplicity, the ability to deploy using existing hardware and software, and the familiarity of most IT organizations with the underlying technology and tools, namely IPSec/PPTP-based VPNs. Specifically, most corporations have implemented and gained

[1]Portions of this work were done when S. Ganu was visiting Avaya Labs.

experience with telecommuter access to corporate networks over IPSec, and adding wireless access is viewed as only an additional profile for an end-user. Most laptops also have IPSec clients already installed to enable telecommuter access. IPSec secures only IP traffic, but given the access patterns of most users, this is currently not a severe restriction. The changes needed for implementing the deployment architecture shown in Figure 1 are only incremental, which is also a significant factor in its favor. While the requirements of 802.11i can be implemented as software upgrades on APs, the general expectation is that performance reasons would necessitate hardware upgrades. The current expectation is also that most enterprises would wait for some time to ensure amortizing the cost of current 802.11 hardware before considering upgrades. Therefore, using VPNs to secure enterprise wireless networks is a viable option and a recommended practice that is and will be used.

While VPN authentication and encryption mechanisms are strong, the current architecture remains vulnerable to attacks if the VPN server can be bypassed. In this paper, we describe and demonstrate how the use of dual interface (i.e., wired and wireless interfaces) machines, such as standard laptops, can compromise wireless security if standard operating system features are intentionally, unintentionally or maliciously activated. This loophole essentially opens up the network to attacks such as "war-driving" [12], the "parking lot scenario" [2], and attacks from unprotected lobbies and floors in companies.

The remainder of this paper is organized as follows. In Section II we present in detail the vulnerability in the deployment architecture from Figure 1. In Section III we outline several solutions to prevent such attacks. The solutions are based on techniques ranging from pure monitoring to active detection. We also present simple software features that, when added to current APs, can help resolve the loopholes. We present some simple experiments and practical observations in Section IV and conclude in Section V.

II. The Hidden Wireless Router (HWR)

Most wireless devices, and in particular, wireless-enabled laptops have dual network interface cards (dual-NICs), e.g., they have an in-built 802.11 chipset and also an Ethernet adapter for obtaining network connectivity. Many enterprises provide both Ethernet jacks and an 802.11- network using the architecture from Figure 1 for providing users with network access within their location. Users may connect to an Ethernet jack, or the wireless network and its VPN server, or both. Usually, the Ethernet jacks are "open" (i.e., require no authentication), but can be enabled for 802.1X authentication.

Implicitly, the architecture from Figure 1 assumes for its security that all wireless clients will access the network through the VPN server, and tries to ensure this by providing users with only a non-routable IP address upon association with an AP. The problem lies in this implicit assumption. Consider the scenario from Figure 2, where a dual-NIC laptop, *H*, is associated with an AP and has obtained a private IP address. However, the user has connected the laptop to an Ethernet jack and obtained a routable IP address on the wired interface. The path the packets take from the laptop to the corporate intranet goes through the Ethernet interface and directly into the Intranet. Now, assume that this laptop has network address

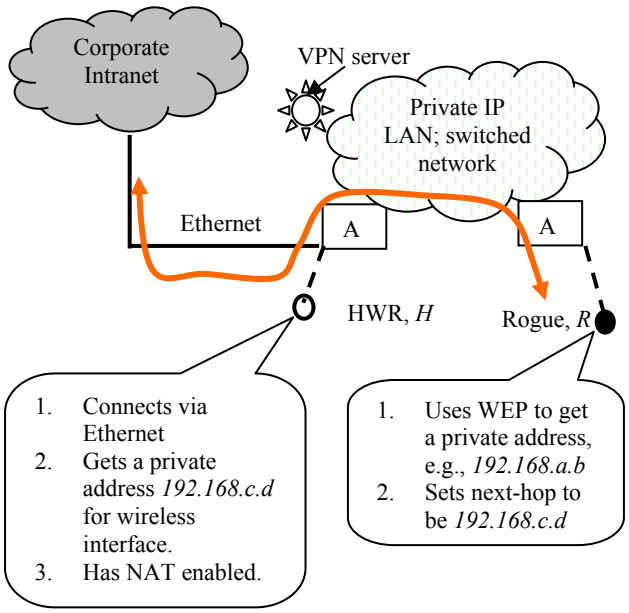

Figure 2. The hidden wireless router vulnerability

translation (NAT) enabled on it. (Later in this section, we elaborate on how this can be easily set up.) Suppose a rogue client, *R*, forwards its packets to *H*. These packets (and all responses) will find their way between *R* and the corporate intranet via *H,* bypassing the VPN server. Since the rogue *R* only needs WEP to associate with an AP and get a private IP address, it has successfully broken into the corporate intranet. We refer to this scenario as the hidden wireless router (or HWR) vulnerability, since the legitimate laptop *H* is acting as a hidden router with or without its knowledge. Notice that since *H* is simultaneously a legitimate client and a NAT router, the hidden wireless router vulnerability does not depend on 802.1X being enabled on the Ethernet jacks.

The possible seriousness of the vulnerability is compounded by the observation that it is rather trivial to enable a dual-NIC laptop to be a NAT router. Many operating systems (e.g., Linux, Windows, etc.) allow users to turn on NAT. Since many end-clients are Windows machines, we remark on how NAT can be enabled to make such a machine a potential hidden wireless router. In Windows (e.g. 2000 and XP versions), connection sharing can be enabled on the wired interface with the wireless interface as the "local network." This automatically sets an address of 192.168.0.1 for the wireless interface. One can then configure the wireless interface to use DHCP. This does not remove "internet connection sharing" and allows the wireless interface to obtain its address from the DHCP server, completing the setup. (Note that re-configuring the wireless interface to use DHCP is not essential. In particular, as long as both the rogue and the machine *H* are associated with an AP, the rogue *R* can assume an IP address in the subnet of client *H*'s private IP address (e.g., an IP address of 192.168.0.5) and route packets to *H* enabling the attack; these packets will be, under normal circumstances, forwarded

from *R* to *H* since the APs and the switches connecting them essentially do Layer-2 forwarding. We observe that it should be possible for machines to get set up in this configuration in a number of ways: getting hacked at public networks, via viruses and worms, a simple misconfiguration by users, not actively adjusting the configuration between home use and corporate use, etc. In particular, the conceptual hidden wireless router problem remains; it is the ease of activating it - potentially without the user's knowledge - that makes this a potential high risk.

We make some observations related to the HWR vulnerability:

- First, viewing only the associations and DHCP activity, one would not see anything unusual in how the rogue *R* and the hidden wireless router *H* are connected to the network; in-built wireless cards in laptops would exhibit precisely such an association behavior if the user has left the card in the appropriate profile.
- Second, the rogue *R* and the client *H* do not need to be within radio range; in particular, the packets from the rogue *R* should get switched to *H* over the corporate intranet switched network connecting the APs. This indicates that the HWR vulnerability is likely more serious than the well-understood rogue access point issue that requires an "insider" to have connected an unauthorized access point[2] to a corporate Ethernet jack.
- Third, we have presented the discussion in the context of two physical interfaces on the laptop, a wired and a wireless interface, when the VPN client is not used (since the laptop is connected through the wired interface). However, the problem does exist even with a purely wireless laptop, depending on the VPN client used for connection. For example, we have observed that enabling connection sharing on the logical PPTP interface with Microsoft's PPTP client will allow packets from the (unprotected) private address wireless interface to get NAT-routed to the VPN tunnel. (Also see Section IV.) However, many VPN clients disallow split tunneling and will reject packets on the unprotected interface, and should, therefore, not pose an issue when the VPN is active (on the wireless interface).

While the focus of this document is on the HWR-scenario, we would like to mention that similar threats arise if bridging or other forwarding mechanisms are used. (An experiment using a bridge will be described in Section IV.) The solutions in Section III can be easily adapted for these other forwarding mechanisms as well.

[2] Note that the rogue access point problem really requires intentional subversion of corporate policies in getting unauthorized hardware connected to the network, as opposed to the HWR vulnerability that can arise from, e.g., software misconfigurations. In fact, one could argue that the rogue access point is more a wired authentication vulnerability as opposed to a wireless vulnerability that should be tackled by authentication mechanisms like 802.1X or its derivates.

III. POSSIBLE SOLUTIONS TO THE HWR PROBLEM

In this section, we will describe different approaches to tackling the HWR problem, namely monitor-based and Access Point-based solutions. While monitor-based solutions are aimed at detecting, locating and controlling HWRs in a reactive manner, Access Point-based solutions are proactive methods that prevent HWRs from operating at all. It should be noted that all monitor-based solutions described here could also be implemented in the Access Points whereas the converse is not true.

As outlined above, the HWR problem stems from two facts. Firstly, when securing wireless local area networks with a VPN, a non-authenticated station may be able to associate with a wireless Access Point. Secondly, a (legitimate) dual-homed machine may be connected to both the wireless network and may forward traffic from the wireless network to the wired network and vice-versa. Clearly, mandating that wireless clients must either not forward traffic or be connected to the wired network as well would solve the problem. Software could be put on clients (e.g., bundled with the VPN clients) to warn users when connection sharing is detected, and such software could also enforce disabling IP packet forwarding on client machines. However, while client-based solutions provide some deterrent, they can very hard to enforce in a foolproof way. Therefore, we will focus our attention on non-client-based solutions.

A. Monitoring-Based Solutions

In this section, we will present solutions based on monitoring the traffic in the wireless network. We will refer to such devices as *sniffers* [13][15]. The sniffers may be passive and purely listen to the air [13], or they may additionally be stations in the wireless network.

1) Detecting HWR

In the HWR scenario, security is compromised since a non-VPN-authenticated station can gain access to the enterprise network bypassing the VPN server. In other words, VPN-protected wireless access is "safe" if all traffic from the wireless network is getting consolidated at the VPN server since, in this case, traffic from non-authenticated stations is dropped. Therefore, monitoring *cross-traffic*, i.e., traffic from a wireless station that is not destined to the VPN server but to another wireless station, is the key to detecting HWRs.

While the cross traffic is similar to *ad hoc* traffic, it does not stand out as such since both stations are operating in the infrastructure mode. Cross-traffic can be easily identified by a sniffer observing only MAC headers of the 802.11-frames. By observing traffic, and communicating amongst themselves, the sniffers can learn the MAC addresses of all connected wireless stations and APs. Cross traffic is all traffic in which the source and destination addresses are wireless stations. Alternatively, the sniffer could be configured with a list of permissible MAC destination addresses such as those of the VPN server or address of the gateway to the VPN server. We conclude that a sniffer can identify cross traffic even if it is not in possession of the WEP-encryption key of the wireless network since the frame headers are transmitted in the clear. While a discussion of whether cross-traffic is useful and should be permitted at all

in VPN-secured wireless networks is beyond the scope of this document, we would like to point out that it is also possible to identify HWR-traffic in observed cross-traffic if the sniffer can WEP-decrypt frames. This can be done using several methods. The sniffers could maintain a mapping between (source/destination) MAC and IP addresses in observed (non-broadcast) frames. A MAC address that occurs with more than a single IP address indicates that a router may be present. If the MAC address is that of a wireless station, an HWR may be present. It is also possible to specify a list of permissible routers and have the sniffer report any non-specified devices that appear to route traffic. Alternatively, if the destination (source) IP address is not in a permissible range (e.g., the private IP address range for the wireless network), the source (destination) MAC can be flagged as a rogue client and the destination (source) MAC as an HWR, if located in the wireless network.

While the approaches discussed above are passive, HWRs can also be detected in an active manner, since unlike wired NAT scenarios, in the HWR case we have access to the "other side of the NAT", namely the wireless medium and can send packets to the HWR on this interface. The sniffer in this case is associated and tries to establish a connection to a "honey pot" server in the wired network using a suspected HWR as the gateway. (In other words, the sniffer acts as a rogue client.) If a response is received, the device used as the gateway is identified as HWR. Furthermore, the "honey pot" can return the IP source address of the received packet. Thus, the IP address of the wired interface of the HWR is known. The devices targeted by the sniffer can be all (active) stations or only the stations that have been marked suspect using the techniques described earlier in this section. Any device in the same subnet could also perform this task. The limitation of probing the private address space is that this approach would not capture HWRs that operate with an IP Address outside of the probed private address range.

2) Locating and Controlling HWRs

After a (potential) HWR has been identified, it is necessary to locate the HWR and to control it. If a monitor-based approach is pursued, the HWR can be located by using location-estimation techniques based on signal-strength measurements, as for instance outlined in [14][15][16]. If an active probing technique as outlined above has been successfully applied, the HWR may also be located from the wired side. By tracking the known IP address of the wired interface of the HWR, the HWR's whereabouts can be traced back to a switch-port/jack-number, which can be resolved to a location. The IP address may also be mapped to a particular user through a database.

The IP address can be used to pop up a message at the machine or the user (if discovered) can be informed. Sometimes, locating the device, informing the (unsuspecting) user about the problem and fixing it from the client side may take some time. In such cases, the HWR should also be immediately disabled by non-client-based methods in order to prevent further misuse. In this case, either the wireless or the wired network connection of the HWR needs to be disabled. For the wireless side, the MAC-based access control feature as implemented in most APs could be used to contain the HWR. If the APs are instructed to add the MAC address of the HWR to a list of not-allowed stations and a *disassociate* message is sent to the HWR, it loses connectivity to the wireless network. From the wired side, the network jack that the HWR connects to may be disabled or routers and switches may be instructed not to forward traffic to this device.

Note that detecting and controlling HWRs is to some extent related to other issues in wireless LAN security like MAC/IP address spoofing and its use in denial of service attacks. While some or the techniques described in this paper may be useful in tackling these problems, the discussion of such issues is beyond the scope of this paper.

B. Access Point-Based Solutions

While the monitor-based solutions provide protection against the HWR scenario, they are based on reacting to a detected HWR as opposed to preventing an HWR from operating. If possible, the AP can prevent the HWR scenario by frame filtering based on MAC source and destination address. As outlined above, the AP may also take into account IP addresses such that this approach can either deny all or still allow cross-traffic. A common objection to AP-based access control lists is that they are hard to manage and are not scalable. We note that this is not the case with our solution. The list of *permissible* addresses is limited to a few entries (e.g., primary and backup VPN servers) and needs to change only if the properties of the VPN server are changed

IV. EXPERIMENTS AND OBSERVATIONS

We performed experiments on two networks, *N1* and *N2* in different locations. The first network *N1* was protected by a PPTP-based VPN mechanism, while the second network *N2* was protected using an IPSec-based VPN mechanism (with split tunneling disabled). Our dual-interface laptop was running the Windows 2000 operating system.

A. Verifying the HWR vulnerability

We connected laptop *H* to the wired network and it also associated with an AP. The laptop *H* got a private IP address on the wireless side and a routable IP address on the wired side, both through DHCP. By enabling connection sharing on the wired interface of *H*, we turned the device into an HWR using the technique outlined in Section II. The rogue station *R* (in our case, a Linux laptop) also associated with an AP and got a private IP address through DHCP. When not activating the VPN client on *H*, the rogue *R* exploited the HWR vulnerability by setting its default gateway address as the IP address of *H*'s wireless interface. This worked in both networks *N1* and *N2* and even when *R* and *H* were associated with different APs.

B. Exploiting Vulnerability Through Bridging

We also used the network bridging feature available in modern operating systems for compromising the security of the wireless VPN-architecture. In this case, bridging was enabled on device *H*, between the wireless and the wired interfaces. The rogue *R* used *H* as a bridge into the enterprise network by using an IP address in the wired IP address space.

C. *Effect of Enabling VPN on the HWR*

In this experiment, we enabled the VPN client on *H*. In network *N2*, enabling the VPN client on *H* disrupted the operation of the HWR. We were also unable to ping *H*'s wireless interface on its non-routable IP address, since all packets to the raw interface were dropped by the VPN client. In network *N1*, activating the PPTP-based VPN did seem to disrupt the operation of the HWR; however, we could still ping the non-routable IP address on *H*. After changing the default route on *H* to point to the default gateway for the wired subnet, *H* again acted as an HWR.

D. *HWR with Single Physical Interface*

Inspired by the previous observation, in *N2,* we disconnected *H* from the wired network and enabled connection sharing between the PPTP interface and the wireless interface. Laptop *H* now had only one physical (wireless) interface active, with two logical interfaces. We found *H* operating as an HWR in this configuration with packets getting NAT-forwarded from the raw wireless interface to the PPTP interface. When the VPN was disabled and a packet was sent from *R*, the laptop *H* even prompted for VPN enablement!

E. *Detection of HWR*

We also experimented with the monitor-based detection techniques for HWRs as described in Section III.A. Specifically, we used the MAC and IP-address based method to detect cross-traffic and HWR-traffic. These methods did find a HWR we placed into the network. We also used the active probing-technique in combination with a "honey pot'' server to detect the HWR. This method did identify the planted HWR and yielded its wired IP address as expected.

F. *Estimating the Extent of the Vulnerability*

In network *N2*, we probed the wireless non-routable address space by sending 2 ping packets each spaced 0.2 seconds apart and with a maximum wait of 1 second to each address. We found that 87 addresses responded. Usually, approximately 160 clients are authenticated with the wireless VPN server. As described earlier, authenticated machines in network *N2* do not respond to a ping to the raw interface. Therefore we conclude that, in this case, approximately 35% of laptops are associated with an access point but not running a VPN client leaving them vulnerable as HWRs, if connection sharing were to be enabled.

We conclude that the HWR vulnerability is indeed a threat to network security for wireless VPN installations that can be detected and controlled by the procedures outlined in this paper.

V. CONCLUSION

Wireless VPNs are being increasingly deployed to secure enterprise wireless networks. In this paper we described the Hidden Wireless Router Vulnerability for VPN-secured wireless local area networks. We presented methods to detect, control and prevent rogue terminals from exploiting this vulnerability. Our experimental results show that the behavior of enterprise users might result in a significant number (35% in our test) of legitimately connected wireless terminals being susceptible to becoming HWRs. Furthermore, the implementation of our monitoring-based approach is suitable to detect and control HWRs.

In the future, we plan to incorporate the HWR vulnerability detection techniques outlined in this paper into our platform for monitoring enterprise wireless networks[15]. This prototype implementation will also allow to locate and to deactivate HWRs from both the wireless and the wired side.

REFERENCES

[1] Intel Centrino, http://www.intel.com/home/notebook/centrino/

[2] W. Arbaugh, N. Shankar, Y.C. Justin Wan, "Your 802.11 Wireless Network has no Clothes," in *Proc. of the First IEEE International Conference on Wireless LANs and Home Networks,* December 2001.

[3] N. Borisov, I. Goldberg, D. Wagner, "Intercepting Mobile Communications: The Insecurity of 802.11," in the Proc. of the 7th Intl. Conf. on Mobile Computing and Networking, July 2001.

[4] S. Fluhrer, I. Mantin, and A. Shamir, "Weaknesses in the Key Scheduling Algorithm of RC4," in *Proc. of the 8th Annual Workshop on Selected Areas in Cryptography,* August 2001.

[5] A. Stubblefield, J. Ioannidis, A. Rubin, "Using the Fluhrer, Martin, and Shamir Attack to Break WEP," in *Proc. of the 2002 Network and Distributed Systems Security Symposium,* February 2002.

[6] Geier, Jim, *802.11 WEP: Concepts and Vulnerability*, 802.11 Planet, June, 2002, http://www.wi-fiplanet.com/tutorials/article.php/1368661.

[7] IEEE 802.11i standard, http:// grouper.ieee.org/groups/802/11/Reports/tgi_update.htm.

[8] IEEE 802.1x Port-Based Network Access Control standard, http://www.ieee802.org/1/pages/802.1x.html.

[9] A. Mishra and W. A. Arbaugh, "An Initial Security Analysis of the IEEE 802.1X Standard," Tech. Rep. CS-TR-4328, University of Maryland, College Park, February 2002.

[10] VPN and WEP: Wireless 802.11 Security in a wireless environment, http://www.intel.com/Ebusiness/pdf/it/wp021306.pdf.

[11] Michigan Eng. CAEN Wireless Network, http://www.engin.umich.edu /caen/ network/wireless.

[12] Wireless WarDriving, http://www.personaltelco.net/index.cgi/War-Driving.

[13] Kismet Wireless Sniffer Homepage, www.kismetwireless.net.

[14] P. Bahl, V. N. Padmanabhan, "RADAR: An In-Building RF-Based User Location and Tracking System," in *Proc. of IEEE Infocom*, March 2000.

[15] S. Ganu, A. S. Krishnakumar, P. Krishnan, "Infrastructure-based location estimation in WLAN networks," in Proceedings of the 2004 IEEE Wireless Communications and Networking Conference, Atlanta, Georgia, March 2004.

[16] P. Krishnan, A. S.Krishnakumar, W. H. Ju, C. Mallows, S. Ganu, "A System for LEASE: Location Estimation Assisted by Stationary Emitters for Indoor RF Wireless Networks," in Proceedings of the 23rd IEEE Conference on Communicaiton (INFOCOM), Hong Kong, March 2004.