A Flexible, Privacy-Preserving Authentication Framework for Ubiquitous Computing Environments*

Jalal Al-Muhtadi Anand Ranganathan Roy Campbell M. Dennis Mickunas
Department of Computer Science,
University of Illinois at Urbana-Champaign,
{almuhtad, ranganat, rhc, mickunas}@uiuc.edu

Abstract

The proliferation of smart gadgets, appliances, mobile devices, PDAs and sensors has enabled the construction of ubiquitous computing environments, transforming regular physical spaces into "Active Information Spaces" augmented with intelligence and enhanced with services. This new exciting computing paradigm promises to revolutionize the way we interact with computers, services, and the surrounding physical spaces, yielding higher productivity and more seamless interaction between users and computing services. However, the deployment of this computing paradigm in real-life is hindered by poor security, particularly, the lack of proper authentication and access control techniques and privacy preserving protocols. We propose an authentication framework that addresses this problem through the use of different wearable and embedded devices. These devices authenticate entities with varied levels of confidence, in a transparent, convenient, and private manner, allowing the framework to blend nicely into ubiquitous computing environments.

Keywords

Ubiquitous computing, security, authentication, context-awareness, privacy, Mist

1. Introduction

Ubiquitous computing or *Active Information Spaces* promote the proliferation of embedded devices, smart gadgets, sensors and actuators. We envision an Active Information Space to contain hundreds, or even thousands, of devices and sensors that will be everywhere, performing regular tasks, providing new functionality, bridging the virtual and physical worlds, and allowing people to communicate more effectively and interact seamlessly with available computing resources and the surrounding physical environment. This vision of Active Information Spaces is not far fetched; the Gaia project [1][2][3] at the Department of Computer Science, University of Illinois at Urbana-Champaign, attempts to develop a component-based, middleware system that provides support for building, registering and managing applications that run in the context of Active Information Spaces. However, the real-life deployment of Active Information Spaces is hindered by poor and inadequate security measures, particularly, authentication and access control techniques. Active Information Spaces promote the automation of some services (e.g. automatic adjustments of lighting and air conditioning), and the anytime, anywhere access to resources, in an attempt to enhance users' productivity and services' availability. However, these same features give enormous leverage to cyber-attackers, hackers, and unauthorized intruders allowing them to inflict greater damage once they break into the system. Also, Active Spaces encompass both the virtual and physical worlds; this makes them prone to more severe security threats and vulnerabilities that could threaten people in the physical world besides threatening their data and programs in the virtual world.

Most traditional authentication methods either do not scale well in massively distributed environments, with hundreds or thousands of embedded devices like Active Spaces, or they are inconvenient for users roaming around within Active Space environments. Moreover, authentication in Active Spaces cannot use a "one-size-fits-all" approach, as authentication requirements differ greatly among different Active Spaces and different applications and contexts within the same Active Space.

Different applications have highly varied authentication requirements. Some like a weather service may be accessible by anybody. Other services, like controlling a power grid may require a person to be authenticated with a "high-level" of "confidence." This may require him to pass various checks like fingerprint recognition, retinal scan, face recognition, remembering a password, etc. We need a model that can handle this range of authentication requirements.

In this paper we propose an authentication framework that provides a flexible and convenient authentication and

* This research is supported by a grant from the National Science Foundation, NSF CCR 0086094 ITR and NSF 99-72884 EQ.

access control services for Active Spaces. The framework's flexibility is demonstrated through its ability to support multiple authentication devices and methods, while allowing new authentication technologies to be incorporated dynamically. The framework enables the use of different wearable and embedded devices to authenticate entities with different levels of confidence. However, the use of wearable devices and active badges could severely violate the location privacy of users. Without careful design, such a system can become an effective surveillance system. We employ Gaia's Mist communication protocol [5][6] to authenticate users while preserving their location privacy.

This framework is capable of scaling to massively distributed systems, while supporting the dynamism and flexibility that Active Spaces promote, and being customizable enough to adapt to different privacy and authentication requirements of different Active Spaces and different contexts within a single Active Space.

The remainder of this paper is divided as follows. Section 2 talks about the various authentication devices that we use in our authentication framework. Section 3 shows how our system uses confidence values to provide greater flexibility. Section 4 illustrates our authentication protocol. Section 5 briefly mentions how context-sensitive information is incorporated into our framework.

2. Authentication Devices

In this section, we briefly describe the authentication devices that are incorporated in our Active Information spaces. We examine their capabilities and reliability.

2.1 Active Badges

In our environment, each person has an RF-based active badge that can transmit identification information [9]. This identification information is in the form of a 32 byte string. This string can be written into the badge. The transmitted ID is received by base stations that are positioned in different locations. The base stations can detect badges within a range of 3-20 ft. This range can be set according to the requirements of the system. Badges can thus give the location of a person in terms of which room he is in (although the RF signals can penetrate walls sometimes and give wrong information). Badges can also give the location at a sub-room granularity if there are a number of base stations in different parts of the room and their ranges are set appropriately. On their own, these badges are not a very reliable means of authentication. This is because the badges transmit the identification number in plaintext and this can be easily captured and replayed by someone else. Also badges can be lost, stolen or left behind somewhere. Further, these active badges have limited processing and storage capabilities. However, the usage of active badges does not require any sort of intervention on the part of the user since they keep transmitting all the time and can, hence, be continuously detected. So, we use active badges as a way of finding out where exactly a user is inside a room.

2.2 Smart Jewelry

Jewelry can be worn at all times, is harder to steal and does not require a user to carry additional gear. Therefore, computerized jewelry can provide a convenient way for authentication. We use the iButton® [7] as a prototype for this kind of devices. The iButton is a 16mm computer chip armored in a stainless steel can. It allows up-to-date information to travel with a person or object. The steel button is rugged enough to withstand harsh outdoor environments. The Java powered iButton has a microprocessor with a JVM running inside it. It also has support for performing cryptographic operations. Special ports allow a user to plug his ring into them. The iButton can then exchange information with a computer. If each user has a ring, it can function as a means of authentication. The ring can store a users name and password encrypted with a key only known to the authentication server.

2.3 Smart Watches

Another wearable device that is worn by people almost in daily basis is wristwatches. A "smart" watch can be used as an interactive wearable device, providing a higher degree of secure authentication. In contrast to the iButton, a smart watch stores more information, packs more processing power, features a display, and enables a user to interact with the device. These capabilities make a smart watch a more secure authentication device.

For our system we use the Matsucom's OnHand™ PC wristwatch [8] which packs a 16-bit microcontroller running at 3.64 MHz, 2 MB of flash memory, 128 KB RAM, and an LCD.

2.4 PDAs

In addition to the wearable gadgets, larger PDAs are also used for authentication purposes. These include J2ME-enabled mobile phones, which run a lightweight version of Java (J2ME), Compaq iPAQs and HP Jornadas which run Windows CE™. These devices feature much more processing power (ranging from 16 MHz to 206 MHz) and storage capacity. While PDAs can be lost or stolen more easily than wearable gadgets, their processing, storage and

Figure 1: Some Authentication Gadgets

interactive displays can be utilized to provide better authentication.

2.5 Passwords

Traditional authentication through username and password pairs can be handy when a user does not have access to other authentication devices, or as an additional authentication mechanism that can leverage other authentication mechanisms by drawing on the target's knowledge of some secret information. However, to meet our privacy goals, instead of using an actual username and password pair, we use a pseudonym and password pair. This prevents the client machines from positively identifying the user. Only the authentication server knows the actual mapping between the pseudonym and the actual username. Users can change their pseudonyms for increased privacy.

2.6 Biometrics

Biometrics can be used as an effective mean of authentication. They authenticate users based on their unique physical characteristics, so that users are identified based on "what they are." This may include fingerprints, retina, and voice or face recognition.

3. Multiple Levels of Authentication with "Confidence" values

In a ubiquitous computing environment, users can, as we have just seen, authenticate themselves to the system using a variety of means. In such a scenario, some means of authentication are more reliable than others. For example, it is not difficult to steal someone else's badge and walk into different rooms with it. Passwords can also be cracked by simple guessing or using brute force algorithms. Fingerprint identification is a fairly good means of authentication. Therefore, we need a model that captures the fact that not all authentication methods are indistinguishable; rather, some may provide significantly stronger authentication than others.

A person in a ubiquitous computing environment can choose to authenticate himself using any one of the available means. He could even use multiple means of authentication. To capture all this, our system assigns different confidence values to different authentication methods. These confidence values give a measure of how "confident" the system is that the person, who has just authenticated himself using some particular means, is indeed who he claims himself to be. For example, we have given a confidence value of 0.6 to authentication using an active badge. This means that when a person, say Bob, has authenticated himself using an active badge, then the system's "confidence level" that the person is really Bob is 0.6. It is possible that someone else has stolen or reproduced Bob's badge, or that Bob has left his badge in his office and his authentication has taken place in the wrong room. Authentication using fingerprints has been given a confidence value of 0.95. This also implies that fingerprint authentication is more secure than authentication using an active badge.

When a person uses more than one authentication method, then the overall level of confidence increases. In this case, we introduce a *confidence-builder module.* This module employs some algorithm for combining multiple confidence values in some manner, and producing a net confidence value. We implement this as a module to enable us to plug-in different algorithms for combining and "reasoning" about the confidence values. In our current implementation, we employ a simple probability-based formula for calculating the net confidence value:

$c_{net} = 1 - (1-c_1)(1-c_2)\ldots(1-c_n)$

Where c_{net} is the net confidence value of a person who has authenticated himself using n methods whose individual confidence values are c_1, c_2,.., c_n. The intuition behind this is that $(1-c_i)$ represents the "probability" that the person was incorrectly authenticated by method i. The product of all $(1-c_i)$ terms gives the probability that the person was incorrectly authenticated by all the methods he used. So, finally c_{net} gives the "probability" that this did not happen. For example, if a person authenticated himself using a badge and his fingerprint, then the net confidence value is 1 - (1-0.6) (1-0.95) or 0.98. We plan to investigate the use of other algorithms for combining confidence values, like Bayesian probability or fuzzy logic.

This notion of different confidence levels of authentication can be used by applications or services in access control decisions. Certain highly-secure services can choose to only serve those clients who are authenticated with a relatively high confidence. For example, starting or stopping certain core services like the discovery and naming services in Gaia, or the printing service so that the correct person is billed. However, a jukebox application might decide to be accessible to users with lower confidence values. Accordingly, if a person wishes to use some not-so-critical applications he can authenticate using just his badge. However, if he wants to access more secure applications, he needs to authenticate himself using different methods.

4. Authentication Protocol

4.1 Limitations of Existing Protocols

While Kerberos [4] was a success in meeting authentication challenges in early distributed systems, it has serious limitations that hinder its effectiveness in ubiquitous computing environments. First, it is mainly based on passwords, and as such is prone to password-guessing attacks. Second, Kerberos assumes that every user in the system accesses services through a designated workstation. In other words, a user has to log into some workstation on the network and only from that workstation the user can access the distributed services. On the contrary, in a ubiq-

uitous computing environment there is no notion of a "single machine" that the user uses to access the available services. Instead, the user can access the services through any of the hundreds of machines that populate the Active Space. Further, Kerberos assumes that the client machines are trustworthy, allowing them to store and use users' tickets. Obviously, Kerberos was never designed to take user privacy into consideration. To meet the challenges of authentication in a ubiquitous computing environment, we propose an authentication framework that resembles Kerberos, but avoids its limitations and scales to physical spaces while taking context and location information into account.

4.2 Privacy Concerns

The use of wearable devices and cloth articles to detect users and authenticate them provides flexibility and convenience; however, the location privacy of users is severely violated. Without careful design, such a system can become an effective surveillance system. To avoid this, some approaches [11] employ a different method for location detection, in which the Active Space broadcasts location information that clients can receive and determine their location with. Although this approach does not require users to reveal their location or identity, such a system greatly limits the actions users can do with the acquired location information if they do not transmit anything to the environment. We envision an Active Space to be able to actively detect the presence of users and objects, and exchange information with them for authentication purposes. We consider these features necessary to make spaces active and enable context-based applications. Therefore, we need a method that allows users to authenticate themselves to the surrounding environment while preserving their privacy.

In Gaia, we introduced Mist [5][6] a communication infrastructure that preserves location privacy in ubiquitous computing environments, while allowing entities to be authenticated at the same time. Here, we just give a brief overview on how Mist works. Mist consists of a privacy-preserving hierarchy of *Mist Routers* that form an overlay network. This overlay network allows users to communicate privately. The Mist Routers route communication packets using a hop-by-hop, handle-based routing protocol with limited public-key cryptography, thus, making communication untraceable by eavesdroppers and untrusted middleboxes. Mist introduces "*Portals*" that are installed in Active Spaces. Portals are devices capable of detecting the presence of people and objects through the use of base stations or sensors; however, they are incapable of positively identifying the users. For example, the portals can be made unaware of the user-badge ID assignments. The positive identification and actual authentication of a user take place at a "higher level" in the hierarchy, high enough not to be able to deduce the actual physical location of the user. More specifically, it takes place at a special Mist Router referred to as a "*Lighthouse*." Only a Lighthouse is able to positively identify and successfully authenticate the user. However, the Lighthouse is kept in the "dark" about the actual physical location of the user (thanks to the hop-by-hop routing protocol). The term Lighthouse is coined, because this special Mist Router somewhat resembles a conventional "lighthouse" that sends out signals to aid in marine navigation, particularly in "foggy" nights. To illustrate, in Figure 2, Alice, who is in Active Space 3, is detected by the Portal in that space. The Portal only detects Alice's badge ID (or other information embedded into other devices that Alice is carrying or wearing) however, this information alone is insufficient to indicate that this is actually Alice. The CS Building Mist Router is designated as Alice's Lighthouse. A secure channel between Alice devices and her Lighthouse is established, going through the Portal, node 1, node 2, and finally node 3. Encryption is employed to prevent private information from leaking. Instead of having a traceable source and destination addresses, packets over this secure link are routed through the use of handles that are valid only over a single hop. The intermediate nodes translate an incoming handle to an

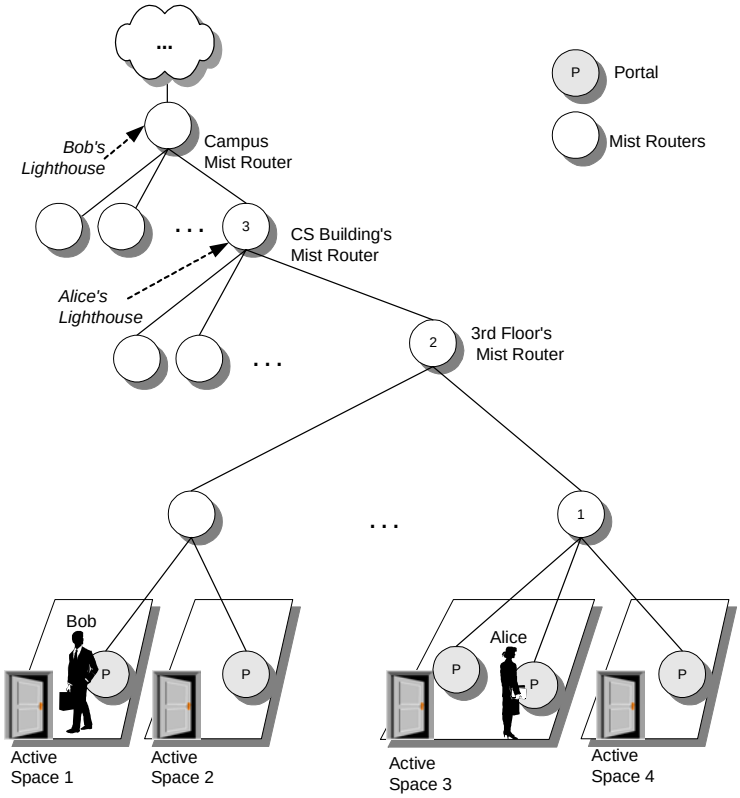

Figure 2: The Mist Communication Protocol

outgoing one. Thus, intermediate Mist Routers can only route to the next hop correctly, but do not know the actual destination or source. Only if all intermediate Mist Routers collude, can the true location of Alice be found. Note that in the example, Alice's Lighthouse can only infer that Alice is located somewhere within the CS building.

4.3 Authentication Protocol

Our authentication protocol extends Kerberos to support user devices and utilize the location privacy that Mist provides. The protocol is illustrated in Figure 3. We give a brief overview of the protocol due to space limitations. Within every Active Space, we assume the existence of one or more "*Space Authentication Portals*" (SAPs). These are special types of Portals that can be located at the entrance of an Active Space, or other convenient places. The SAP will feature a collection of wireless and wired base stations and device readers that enable users to authenticate with the Active Space using any authentication devices they are carrying or wearing.

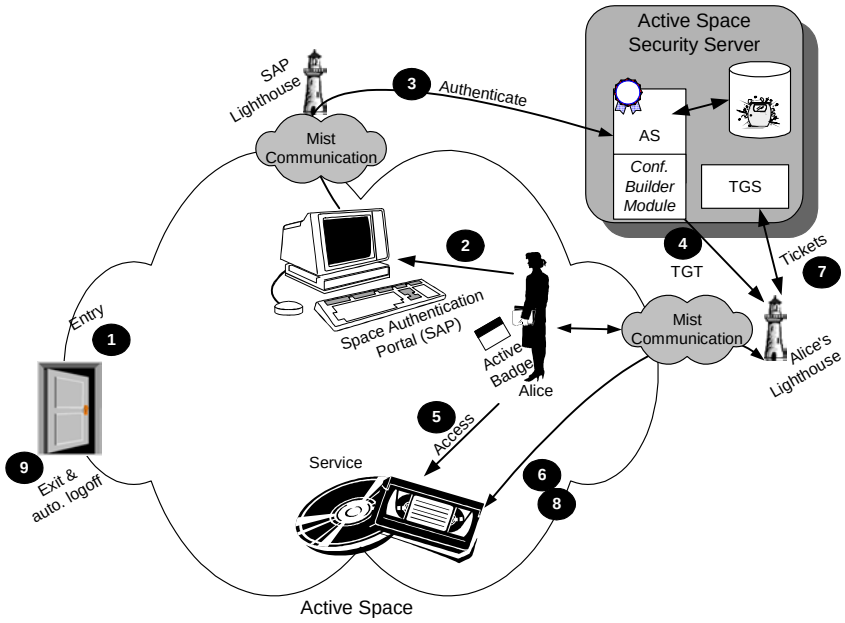

Figure 3: Active Space Authentication Protocol

An Active Space Security Server exists for every *Active Domain*. An Active Domain is a collection of Active Spaces, and the interconnecting networks, which are managed by a single administrative authority. These domains resemble Kerberos "Realms." Like Kerberos, the Security Server consists of three components. The first component is the AS (Authentication Server), which provides a single sign-on point for the Active Domain, using any devices and gadgets the user currently possesses. The TGS (Ticket Granting Server) issues "tickets" that can be used by the user to access available services in that space. These tickets are signed, as a protection against tampering, using the private key of the TGS. Finally, a database is maintained that contains necessary information for the authentication of all users within the Active Domain, as well as their privileges and security attributes.

We assume all users own active badges. More information about these active badges has been given in Section 2.1. Here, we assume that a user badge is programmed to have a unique ID number (by which the AS can identify the user), and an ID to identify the Lighthouse of that user. Note that information embedded in these badges can be updated. Thus, for increased security, the unique IDs identifying users can be changed periodically by the AS and updated into the badges. The badge itself will act as a handle that links the holder to his acquired tickets. Entrances to the Active Space can contain a base station for detecting entering badges (step 1 in Figure 3). This can be useful for services that require users to be physically present, e.g. a printer that only allows people in same space to print.

In step 2, the user moves sufficiently close to one of the available SAPs for authentication purposes. Some authentication devices may require the intervention of the user, e.g. inserting the iButton into its designated receptor.

To achieve privacy, the SAP itself does not have sufficient information to authenticate users. However, it has a Lighthouse through which it can communicate with the Security Server (step 3). Mist communication is used here to prevent the Security Server from pinpointing the exact physical location of the authenticated user. Through its Lighthouse, the SAP contacts the Security Server with a set of authentication requests, each representing a different authentication device. Upon successful authentication, the AS, like Kerberos, issues a ticket granting ticket (TGT) for that user (step 4). Recall that in Mist, every user has a Lighthouse that stores his relevant information. The TGT issued for a user will be stored in his Lighthouse. The TGT in our system is a cryptographic data structure that incorporates a confidence level that is calculated based upon the method(s) used for authentication. The AS remembers which methods the person has used so far to authenticate himself and can, hence, calculate the net confidence of the person being there. Every time a new TGT is issued, the correct net confidence is calculated and stored within the TGT. The TGT will also have an expiration time. Because different authentication methods may have different time-out values, the net confidence may change with time. Therefore, if a TGT expires, the user's Lighthouse may request another TGT from the AS on behalf of the user. If one authentication method fails (for example, if someone used an invalid iButton or en-

tered a wrong password), then the authentication server does not take any action – it does not send a new TGT and the confidence levels of people being in the room are unchanged.

Now, users can access services available in the space without the need to use a "fixed" workstation. Instead, they can interact with the services directly using any capable device (step 5). Upon accessing a secure service, the service will check the user's badge and get the user's ID and the ID of his Lighthouse. The Lighthouse of the user is then contacted by the service. This communication takes place using the Mist protocol to prevent the Lighthouse from deducing the user's location (step 6). In step 7, using the TGT stored at the Lighthouse for the target user, the Lighthouse can communicate with the TGS requesting tickets to access the required service. The TGS issues the necessary cryptographic tickets. These tickets do not contain any references to the real name or identity of the owner; they just incorporate an unforgeable pseudonym. Further, these tickets contain the security privileges of the owner, and the net confidence level. Using the information in these tickets along with the net confidence contained within, the service can make a decision whether to authorize the badge holder or not (step 8). Once a user leaves the room the badge reader at the exits can detect that, automatically logging off the user and destroying the tickets stored in his Lighthouse that are associated with his badge.

5. Context-Sensitive Authentication

The confidence values associated with authentications can be combined with available context information allowing more flexible access policies to be specified. For example, if a person is alone in a room, then access to the PowerPoint service (which allows a user to display PowerPoint slides on a variety of displays) is allowed even with a low confidence authentication. However, if a meeting is taking place in the room, then only the person scheduled to make the presentation is allowed to use the PowerPoint service and this person has to be authenticated with a relatively high confidence.

This context-aware access control makes use of the context framework we have developed for Gaia [10]. The context framework includes a variety of sensors that sense the current situation in a physical space. Sensors send events whenever they detect a change in context. Applications can listen to such context events to capture the current context. They can also query the sensors for specific information. We also have mechanisms to infer more abstract contexts from basic contexts that are sensed. So, it is possible to infer that a meeting is going on in a room if the number of people in the room is greater than, say 5, and the schedule for the room indicates that a meeting is planned in the room.

6. Conclusion

We have presented an authentication framework that builds over Kerberos and introduces new enhancements that allow it to blend nicely into ubiquitous computing environments. The authentication framework enables single sign-on using any devices the user may be carrying or wearing at any time. It allows the decoupling of users from devices and captures some of the dynamism and programmability of Active Spaces by assigning confidence levels to different authentication methods and incorporating context sensitive information. The framework also preserves location privacy for authenticated users. We have employed this authentication framework in the Gaia research project.

7. References

[1] The Gaia Homepage, http://choices.cs.uiuc.edu/ActiveSpaces/index.html

[2] Renato Cerqueira, Christopher K. Hess, Manuel Román, Roy H. Campbell, "Gaia: A Development Infrastructure for Active Spaces," *Workshop on Application Models and Programming Tools for Ubiquitous Computing* (held in conjunction with the UBICOMP 2001), September 2001, Atlanta, USA.

[3] M. Roman and R. Campbell, "GAIA: Enabling Active Spaces," *9th ACM SIGOPS European Workshop*, September 17th-20th, 2000, Kolding, Denmark.

[4] B. Neumann and T. Ts'o, "Kerberos: An Authentication Service for Computer Networks," *IEEE Communications Magazine*, 32(9): 33-38, September 1994.

[5] J. Al-Muhtadi, R. Campbell, A. Kapadia, M. D. Mickunas, and S. Yi, "Routing through the Mist: Privacy Preserving Communication in Ubiquitous Computing Environments," to appear in the Proceedings of ICDCS '02.

[6] J. Al-Muhtadi, R. Campbell, A. Kapadia, M. D. Mickunas, and S. Yi, "Routing through The Mist: Design and Implementation," UIUC Technical Report UIUCDCS-R-2002-2267.

[7] iButton: http://www.iButton.com

[8] On Hand PC Homepage, http://matsucomusa.com/

[9] Active Badges: http://www.rfideas.com/Solutions/Developers/Dev_Kit/Long_Range_/long_range_.html

[10] Anand Ranganathan, Roy Campbell, "A clausal model of context and an infrastructure for context-aware ubiquitous computing," Submitted to Pervasive Computing 2002.

[11] N. Priyantha, Anit Chakraborty, and Hari Balakrishnan, "The Cricket Location-Support System," *Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (ACM MOBICOM)*, Boston, MA, August 2000.