User Access of Public Shared Devices in Pervasive Computing Environments

David Jea, Ian Yap, and Mani Srivastava
Electrical Engineering Department, University of California, Los Angeles

***Abstract*-- To allow for an efficient usage of a device in pervasive computing environments when a user intends to selectively utilize multiple devices within his/her vicinity, reliable and yet convenient authentication is an important requirement. The problem becomes more complex when the accessed device is shared by the public with many different individuals. This paper first illustrates the issues of establishing sessions to such devices (logging in, maintaining a session, and logging out), and then identifies the common pitfall of access-control contexts. We propose an improved context-aware solution that supports a secure, selective, and identifiable user access of public shared devices with high usability.**

***Index Terms*—Body Sensor Networks, Pervasive Computing, User Access**

I. Introduction

People have long envisioned a pervasive computing environment where numerous ambient sensors and actuators are embedded to provide rich experiences of intelligence and interaction. Recent years, Body Sensor Networks (BSN) have employed a similar concept to the human health arena and utilized miniaturized wearable and portable sensor systems to provide remote pervasive health-care monitoring. We imagine in the future that most ambient devices will be shared by the public and can temporarily participate in a user's BSN to provide specific services or data. Such public devices serve as external instruments and are accessible to individuals and can provide private information to authorized users only. Wang et al. [11] suggest four requirements for a secure and flexible access control architecture in ubiquitous computing environments: (1) Access based on contextual information. (2) Easy-to-use access for different users and devices. (3) Collaborative environments. (4)

Decentralized administration. This work investigates the first two requirements by looking into different stages of a session.

Basically, there are three main stages involved when a user is trying to access a device. The first action is to log in successfully (authentication) to use the device. The second stage is to maintain a usage session (presence) with the device. The last action is logging out of the device, i.e. a detachment of use. The usage session with a detached device will be invalidated. We believe that it is important, when a user is trying to access multiple public devices among all other devices that can possibly be used by the public, to address all three stages from four aspects: ***security, selectivity, usability,*** and ***identity***. To access these prevalent public shared devices that can potentially convey private information, the security has been fully investigated in literature. However, considering environments where multiple users are surrounded by pervasive public shared devices, the numerous existing schemes have not fully addressed the remaining three aspects reasonably (Table 1). For example, the traditional authentication techniques such as using account numbers and passwords are straightforward but constitute a substantial usability obstacle to accessing future pervasive computing environments effectively.

A main contribution of this work is that we have identified in the user access methods from previous works the common pitfall is that ***"the context to gain access control of a device is not necessarily the same context to maintain a session and/or the context to release the access control. Moreover, the fact that someone is connected to a device is insufficiently to describe whether the same user is using it."*** To explain this issue, we first present the related work in access control in section II, and then discuss the problems of these techniques in section III. We propose in

section IV a context-based user access solution and describe our system design in section V. Section VI summarizes the paper.

II. Related Work

A complete access control requires a user to authenticate with a device, to employ proper privileges, to establish a secure connection, and to disconnect afterwards. In recent years, the academic world has started looking into other realms of authentication that could be less demanding on the user, and still maintain the security guaranteed by traditional techniques. Stajano et al. proposed Resurrecting Duckling security model in [5] where a device will recognize as its owner the first entity that sends it a secret key (Imprinting). They suggest that imprinting process as a intuitive physical contact. In [4], Balfanz et al. extends the concept in great detail on the use of location-limited channels to be able to authenticate and use a device. The paper provides concrete pre-authentication details and implementation based on infrared. Another similar solution is the Zero-Interaction Authentication (ZIA) protocol from [1]. The idea involves the user being able to authenticate with a computer with minimal interaction. The ZIA protocol considers a user wears a short-range, wireless authentication token to communicate with the system to determine whether to grant access. However, there is nothing mentioned when dealing with multiple laptops.

There is also a proximity-based authentication technique which explores a new domain of authentication: use of context information. These techniques rely on contextual data to authenticate the user and grant usage. Contextual data refer to information that can be gathered in a pervasive environment. Contextual information such as location of the user (from a nearby camera in the ceiling) can help to determine if the user is still in the session. For instance, Bardram et al. [2] describes a hospital in Denmark where the location of a user is provided by an advanced

monitoring system in the hospital. Since it is very cumbersome for users to have to type the id and password multiple times throughout the day to gain access to the database, the context of a user (such as "in the surgery room") facilitates the authentication process when a user tries to use computers near his/her vicinity. Gupta et al. [6] point out that proximity based access control potentially allows someone access the resource, when an authorized user is in proximity. To address the problem, they have introduced three different authentication levels and the notation of proximity zone.

Lately, incorporating context information in user access has received attention in pervasive applications, and different security management models extended from role-based access control (RBAC) [3] has been proposed to explore this issue. The basic concept of RBAC is that users associate themselves with specific roles, which in turn are associated with permissions to resources, and users grant access authorization by being members of roles. Moyer and Ahamad propose a Generalized Role-Based Access Control (GRBAC) in [9] that extends the role concept to subject (user), object (resource), and environment. The expressiveness of GRBAC allows context information (captured in environment roles), such as business hour or CPU load, to be included in an access decision. Zhang and Parashar [10] describe a Dynamic Role Based Access Control model (DRBAC) that improves security by using context information to dynamically make access control decisions. It addresses two requirements in pervasive applications: (1) A user's access privileges binds with the user's context. Context information includes environment (location, time) of the user accessing the resource. (2) The access permissions of a resource binds with its system information (CPU usage, network bandwidth, etc.). Overall, the role based access control maps naturally to large, structured organizations.

III. Problem Statement

The focal question we pose is ***how a user can log in/out of multiple selected devices securely among many others in his/her vicinity and identify him/herself as the person who is using the device, while achieving all of these with little usability issues.*** The problems we are trying to solve here can be better illustrated with a scenario. Imagine that Alice enters a public gym for her daily mile on the treadmill. She selects a camera that monitors the treadmills area, selects an electrocardiogram (ECG) sensor (with new electrodes) to wear then starts using one of the treadmills. These public devices (camera, ECG, treadmill) should able to send her training information or physiological data to her BSN until the exercise is completed and she has left the treadmill.

Figure 1: Questions regarding user access of public shared devices: If there are multiple devices, how to effectively select among these devices? How to maintain sessions to the devices? How to logout these devices when finished? How to keep these sessions secure?

As this paradigm shows in Figure 1, there are many pressing issues to consider. When Alice needs to face multiple publicly accessible devices in her vicinity and only needs to use specific ones within the midst, how can she successfully bind with the devices she intends to use both securely and effectively? How can she also cleanly finish up the usage session when she's done and ready to leave? Moreover, her authentication process or usage session should not disrupt or be disrupted by that of someone next to her who might also be using another treadmill too. Considering the

scenario in Figure 2, we expect a public device to be shared by the others, but how can that device verify that the authenticated user, Alice, is the same person who is using it? The purpose of this work is to solve these issues without incurring too much overhead and usability issues. In the following paragraphs, we will first classify the existing techniques into three categories and then elucidate on why they do not make a good solution, before proposing our own.

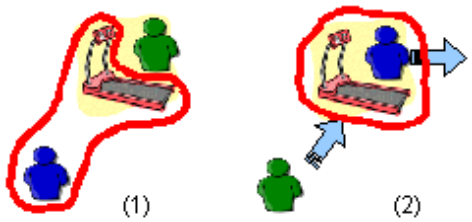

Figure 2: Issues of Identity Confusion: (1) Blue User gains access to the treadmill and the treadmill sends Green User's data to Blue User (2) Blue User is done with the treadmill and leaves. How does the treadmill figure out that the Blue User is gone and identify when the Green User is the new occupant?

For a user to initiate, maintain, or terminate a session with a device, the current state-of-art user access methods can be fundamentally categorized into common wireless technologies (such as GSM, 802.11, Bluetooth, etc.) and location-limited channels (such as physical contact, infrared, short-range RFID, etc.) that exploit the communication medium. The first group is the use of conventional wireless protocols such as 802.11, Zigbee, and Bluetooth. The second group utilizes location-limited channels that are defined in [4] and possess the property where a user can *precisely* control which devices s/he is interacting with. For both categories, a complete session relies on the encrypted information flows on these channels. Additionally, we add a third category that involves user access with the aid of context-based information. We

treat this as a separate category due to its intrinsic uniqueness of utilizing context-based information in a session.

Table 1 lists the caparison of different user access methods in the logging in, maintaining a session, and logging out phases. We discuss in detail of each user access method in the following sections to elaborate why a single solution does not fit in pervasive computing environments.

	Existing Wireless Technologies	Location-Limited Channels	Context-Based
Logging In	Low Usability, especially when logging into multiple nodes	Moderate Usability, user must visit nodes one by one	No selectivity among nodes
Maintaining a session	Without explicitly logging out the nodes, a possible identity confusion problem	Either only one node can be used at a time, or the allowed proximity is rather constrained	To continuously detect the context
Logging Out	Receives a command or/and lost signal	Automatically logged out when heartbeat signal lost	Automatically logged out when the context changes

Table 1: Comparison of existing user access techniques.

Furthermore, role-based access control focuses on issues of permissions-role assignment and is generally a layer higher than the access methods discussed here. However, both GRBAC and DRBAC models employ a similar concept to this paper that context information restrains a user's access to a resource in a session. We thus discussed their similarities and differences to this work in a separated subsection.

3.1 Conventional Wireless Channels

Initiating a session through general wireless channels has low usability in pervasive environments. The main issue here is that it is not easy to dynamically set up a connection between devices on the go without much user interaction. Imagine a user, Alice, with three Bluetooth keyboards in front of her. All three keyboards appear on her PDA to be selected for use. Now, with the conventional wireless channel protocol, Alice is confused on mapping id to the keyboard and getting her PDA to figure out which keyboard to be used. It is apparent that we do not want to employ such awkward processes in future device-rich environments.

Moreover, as shown in Figure 2, maintaining sessions to the authenticated public shared devices through wireless channels could lead to identity issues. If the user does not explicitly log out of a device when he or she is done using it, the device needs to detect that itself explicitly (usually done using timeouts). Imagine, in our scenario, where Alice finishes her daily mile and steps off the treadmill. The next person, Bob, who was waiting in line, now steps up the treadmill and starts using it. Alice will keep receiving Bob's information unless she is out the communication range. There is no way for the system to figure out who is using the device if the existing connection to Alice is maintained through mere wireless channels without any more intelligent observations. This is due to conventional wireless solutions do not seek to exploit the use of “context information”.

3.2 Location-Limited Channels

A user can easily initiate a session to a device through location-limited channel (such as physical contact). The location-limited channel is one extreme case of context-based authentication where a user can precisely control which device s/he is interacting with. Unlike initiation through conventional wireless channels, this

intuitive selection process avoids confusion. We consider that it has some slight usability issues because a user still has to visit the public shared devices that s/he wants to use one by one. For a user to "precisely" select the device, these location-limited channels are either super-short-range (physical contact) or directional (infrared). Therefore, maintaining sessions through these types of channel limits the number of or the distance between connected public shared devices. Such usability issues of pervasive devices constitute an impetus to creating practically useful pervasive computing environments.

3.3 Context-Based User Access

In this type approach, a user will access a device if s/he is in a specific context. Proximity-based access control methods are the most popular methods to determine the context of a user and accessibility of a device. Some example contexts in these methods are "the user is inside the room" or "the user is within proximity zone of a device". The session between an authorized user and a connected device is terminated when the defined context changes (such as when the user leaves the room).

Context-based methods address a more general and flexible way in user authentication and are not limited to proximity-based access control. One example is that, in a station, if Alice faints suddenly (the context), her BSN will automatically authenticate with all nearby defibrillators to grant her access. Also, it will authenticate with the public address system and broadcast emergency messages for a doctor.

However, in a scenario of multiple users in the same context (such as they all in the same room) to unoccupied devices [6], depends on the algorithm, these devices are randomly assigned to all users or exclusively to one user. This appears to be a problem in pervasive environments. In our example, Alice only needs one treadmill, the one she selected, at a time, and neither needs nor wants the neighbor treadmills to

become part of her BSN. Furthermore, defining the proximity zone of each device is also a challenging task. A small proximity zone causes the same issues we discussed for location-limited channels that either few devices or only one device can be accessed by the user at any one time. A large proximity zone (room) has security risks that it potentially allows someone to access a device if an authorized user is within the proximity but not current using it. Without special considerations, a context-based method generally does not select among devices for use.

3.4 Role-Based Access Control

We consider accessing public shared devices in pervasive computing environments as a rather simplified RBAC model (one subject role and one object role). This is a natural result since the public has no clearly structured organizations and thus hard to define roles. However, as discussed earlier and suggested in DRBAC, context information plays the key factor in pervasive applications. Context information in GRBAC or DRBAC captures the environment for use such as time, location, and system information. However, in order to express a user access such as "receiving data from the connected treadmill only if the user is running", we need to apply context information in a more generalized sense that includes the physical state of a subject (user) and an object (device) to describe the usage relationship.

All of these issues constitute the crux of the problem that we want to solve here. We believe that a complete session for a user to access a public shared device within pervasive computing environment should be studied from the different standpoints

IV. Proposed Solutions

The common pitfall in the user access methods is that **the context to gain access control of a device is not necessarily the context to maintain a session and/or the**

context to release the access control. Moreover, the fact that someone is connected to a device is insufficiently to describe whether the same user is using it. Our proposed solution will be as follows. When a user plans to use a device, the device is bound to the user when s/he has established a connection through **initiation-context** (for example, a location-limited channel). When he/she is successfully authenticated with a device and begins to use it, there should be two context states bound to this session, namely a **session-context** (e.g., user leaves the room) and a **govern-context** (e.g., user is on the treadmill). The session-context defines a specified context state of the connection to this session. The govern-context defines a specified state of the user to this device. A device is connected by the user only when the session-context is valid. Based on formalized and customized rules of how the specific context state changes, the device will determine whether the user has logged out and finished using the system. This will then lead to the system cleaning up this current session. As indicated earlier, the session-context is insufficient to describe whether the user is physically using the device. We thus require govern-contexts to represent this relation. A corollary is that a session-context consists of zero, one or more govern-contexts entities that are collectively used to describe a particular session for the user using a particular device. Although most user access methods treat these contexts as one single session-context, a careful design for pervasive applications shall distinguish them. In addition to select proper contexts from usability perspective, the initiation-context ought to provide selectivity among devices and ability to establish secure channels with selected devices, also, the govern-context should provide a continuously monitoring of the physical identity of current user.

V. System Design

As shown in Table 1, there is no single technique addresses all the issues. Thus, a better design incorporates the strength of these techniques. We present the design of a prototype system to demonstrate the solution described earlier. For security related issues, we choose password-authenticated key exchange methods (PAKE) methods to authenticate trust entities and to establish secure communication channels. The user relies on location-limited channels to precisely select a device and use zero-interaction authentication techniques to enhance usability while minimizing user interaction. We expect a scenario where a user exercises in gym that has public sensing devices installed on the fitness equipments. A device joins a body sensor network through the authentication process, but provides sensing information only when the user is indeed physically using the device. To solve the identity confusion problems, we attach one accelerometer to the user and one accelerometer to the elliptical trainer. The elliptical trainer has another SpO2 sensor to monitor heart rate information. This sensor also connects to the user's BSN but only delivers data to the base station when two accelerometers experience similar phenomenon that have same dominant frequency (context matching). With a different user using the same elliptical trainer, the connected BSN (to the first user) stops receiving data. The purpose of this design is to show that a context-based system is smart enough to identify that if the same user is using the connected device. The result prototype system demonstrates our interpretation to a context-based user access system.

Figure 3 shows the state diagram of the system. Besides the login, session maintenance, logout states, we have added a data collection state to reflect the physical usage relationship between the user and the device. The BSN enters to the login state once it detects the initiation-context. In the login state, the BSN

authenticates with the device bounded by initiation-context and establishes a secure channel based on generated session key. The authenticated device becomes a member of BSN and the system then determines the necessary session-context and govern-context involved between the user and the authenticated device. The BSN continuously maintains the session to the device if it detects a valid session-context. When the govern-context is true (attachment), this means that a user is using the device. Otherwise, the device is detached from the user (detachment). And the user's BSN only takes the data of a connected device when the user is actually using the device. The system terminates a connection when the session-context is invalidated. We divide the connection procedure between a user and a device into four stages and explain them in detail: authentication, context establishment, attachment/detachment, and termination.

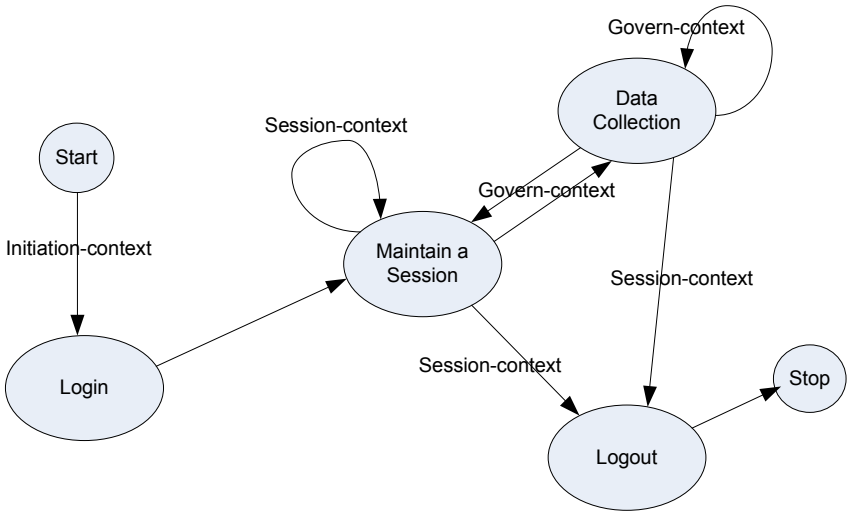

Figure 3: State diagram of the system

5.1 Authentication

Authentication is the first step in establishing a secure communication channel between the user and the device. When using a location-limited channel, a user can easily and precisely select the device s/he wants to authenticate with by first approaching it intuitively. The basic assumption here is that ***at least one location***

limited channel (LED, Irfrad, RFID, sound, physical contact, etc.) is available between a BSN and a public sensor node. After selecting the device, we break down the issue of authentication into two problems that need to be addressed: trust and privacy.

For two parties, the user and the device, to acknowledge and trust each other, we suggest the use of the password-authenticated key exchange methods (PAKE). Encrypted Key Exchange is the first PAKE protocol that published in [7]. The idea is that the user and the device have common knowledge of a secret and when the user authenticates with the device, they challenge each other on the knowledge of the secret without revealing it. The user and the device only trust each other if both parties prove their knowledge of the secret. To reduce security risks, we suggest a use of hybrid keying model where each BSN has a unique key that is shared by BSN members, and a separate global key to access all public shared devices.

The second major issue is privacy. During the process of verifying trust among two parties, PAKE methods such as EKE protocol automatically generate a unique session key based on Diffie-Hellman key exchange algorithm. The method is capable of producing a key that is non-reproducible and is conducive to the current session of interaction. To prevent external cryptographic attacks, security protocols rely on the produced session key to encrypt/authenticate the exchanged messages and allow the user to communicate securely with a trusted device.

5.2 Context Establishment

Context establishment is the next phase where the user needs to establish some form of connection based on the state of using the device. A govern-context describes that if a user is physically attaching to (using) a device and a session-context is the description of the context for the user's current usage session. After authentication

and connecting to a user (meaning that the overall session-context is valid), the device has joined the user's BSN and become one of the BSN nodes. The BSN then decides from a list of contexts the session-context and govern-context of this connection and continuously monitors the two contexts. The system now enters a state of detachment, and if the govern-context is true and session-context is valid, then the system enters the attachment phase and collects data from the connected device. The connection terminates if the system detects the invalidate session-context. Both session-context and govern-context track specific observed states of the physical controlled objects. There are three main types which we discuss.

The first form of the context is the god view, where a central context server that monitors all entities and their behavior in its controlled environment. A govern-context example is the location of a user and the public shared devices. When a user is in proximity of a device, then the system assumes that the user is attaching to the device. Likewise, there are also many possibilities for a session-context. Examples from our treadmill-usage paradigm could be the camera detecting a user leaving the treadmill, or the user waving goodbye at some interpretation device.

The second case is one based on local decision, the context that is determined by the BSN that a user is wearing. The BSN detects current context of the user and the decision is based on this context. For example, if the inferred context of a user is running and a treadmill has been connected to his/her BSN, then the system considers that the user is attaching to the device.

There is the special case where the context can rely on *context proximity* among devices as epitomized in [7]. Their work proposes a scenario where small devices are connected when two persons shake hands, the two artifacts connect with each other if both experience similar context. We extend similar idea to search matchmaking

context among a user and devices. For example, in our treadmill scenario, a user can only attach to a treadmill if similar accelerometer contexts are found on both entities.

5.3 Attachment and Detachment

Attachment refers to that the user has connected to and is using a particular device (session-context is valid and govern-context is true). Detachment is the scenario where the user is temporarily not using the device (session-context is valid and govern-context is false). A reasonable assumption is that when a user attaches to a sensor, it describes the user is using that sensor and no other people (with less or equal access right) can override that relationship physically or virtually. And the sensor believes that all the sampled information belongs to the attached user (exclusive).

After establishing contexts, the device constantly checks if all the govern-contexts involved are true (attachment). If it is, the attachment state is activated and the BSN treats the device as being used by the user and allows information flow from the device to the user's BSN. However, if any of the govern-context fails the device will be considered detached temporarily from the user and stops taking information from the device.

5.4 Termination

The BSN continuously monitors all session-contexts of each connected device and terminates the connection if any of its session-contexts is invalidated. The invalidation of a session-context could occur actively if receives a user command or detects a specific context. It could also occur passively if the detached state holds on for too long or when communication has broken down between the user and device.

VI. Conclusion

In this paper, we have explored user access issues of public shared devices in pervasive computing environments, and suggested a four-staged solution that is based on initiation-context, session-context, and govern-context.

To illustrate this idea and verify the proposed solution, we are in progress to implement a prototype system that demonstrates a scenario of collecting treadmill training information for a user. Future research will be devoted in developing context models and protocols for all the phases, especially investigating how the session-context and govern-context are established between body sensor networks and public shared devices. We look forward to increased interests in facilitating the convenience of user access of devices in pervasive computing environments.

References

1. M. D. Corner, and B. D. Noble, "Zero-Interaction Authentication," *Proceedings of Eighth Annual International Conference on Mobile Computing and Networking (Mobicom),* 2002, pp. 23-28.
2. J. E. Bardram, R. E. Kjær, and M. Pedersen. "Context-Aware User Authentication – Supporting Proximity-Based Login in Pervasive Computing," *Proceedings of Fifth International Conference on Ubiquitous Computing (Ubicomp),* LNCS 2864, Springer, 2003, pp. 107-123.
3. R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-Based Access Control Models," *IEEE Computer 29(2)*, IEEE Press, 1996, pp. 38-47.
4. D. Balfanz, D. K. Smetters, P. Stewart, and H. C. Wong, "Talking to Strangers: Authentication in Ad-Hoc Wireless Networks," *Proceedings of Network and Distributed System Security Symposium (NDSS),* Internet Society, 2002, pp. 23-35.
5. F. Stajano and R. Anderson, "The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks," *Proceedings of the 7th International Workshop on Security Protocols,* LNCS 1796*,* Springer, 1999, pp. 172-194.
6. S. K. S. Gupta, T. Mukherjee, K. Venkatasubramanian, and T. Taylor, "Proximity Based Access Control in Smart-Emergency Departments," *Proceedings of 4th IEEE Conference on Pervasive Computing Workshops, First Workshop On Ubiquitous & Pervasive Health Care (UbiCare)*, 2006, pp. 512-516.
7. L. E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl, and H. W. Gellersen, "Smart-Its Friends: A Technique for Users to Easily Establish Connections between Artefacts," *Proceedings of Third International Conference on Ubiquitous Computing (Ubicomp),* LNCS 2201, Springer*,* 2001, pp.116-122.
8. S. M. Bellovin and M. Merritt, "Encrypted key exchange: password-based protocols secure against dictionary attacks," *Proceedings of the IEEE Symposium on Security and Privacy,* 1992, pp. 72-84.

9. MJ Moyer, M Ahamad, "Generalized Role-Based Access Control," *Proceedings of the 21st IEEE International Conference on Distributed Computing System*, 2001, pp. 391-398.
10. G. Zhang and M. Parashar, "Context-Aware Dynamic Access Control for Pervasive Applications," *Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS)*, 2004, pp. 219-225.
11. H. Wang, Y. Zhang, J. Cao, "Ubiquitous Computing Environments and Its Usage Access Control," *Proceedings of the 1st international conference on Scalable information systems*, 2006, Article No. 6.