Security and Privacy in the Intelligent Room

by

Rattapoom Tuchinda

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2002

Author .....
Department of Electrical Engineering and Computer Science
May 15, 2002

Certified by .....
Howard Shrobe
Artificial Intelligence Laboratory
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Security and Privacy in the Intelligent Room

by

Rattapoom Tuchinda

Submitted to the Department of Electrical Engineering and Computer Science
on May 15, 2002, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

The vision of intelligent environments where computation will be pervasive is no longer a dream. As computers become smaller, they are embedded in everything around us from clothing to wall. Human will be able to interact with intelligent environments using speech or guestures and control or access devices from afar. Intelligent environments open up many possibilities to developers to creat useful applications to hunans and at the same time raises the concern over the issues of security and privacy. The thesis aims to solve these issues by focusing on access control mechanisms, which decide "who has the right to access what kinds of resource, and under what conditions?" We analyze access ocntrol issues in various environment settings from one user in one space to multiple users in multiple spaces and argue that current access control mechanisms lack necessary features to reflect dynamic situations that happens in intelligent environnments. We then present the implementation of Intelligent Access Control that uses a novel approach by separating access control issues into three aspects: security, privacy, and quality of service.

Thesis Supervisor: Howard Shrobe
Title: Artificial Intelligence Laboratory

Acknowledgments

I would like to thank Dr. Howard Shrobe, my thesis supervisor, for giving me freedom to explore the subject and guiding me with any difficulties I might have. I would also like to thank Stephen for his expertise in Latex, Andy for the cookies and sodas, and Krzysztof for encouragement and comments on my thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

As the size of computers become smaller and their speeds become faster, they will be embedded in every device, appliance, or even clothing. MIT's Project Oxygen envisions that computation will essentially be free. E21, a subsidiary project of Project Oxygen, aims at embedding computers into an environment and using their computations to aid humans in their everyday lives. We call this an intelligent environment. An intelligent environment can be any space, such as a meeting room, a living room, or a bathroom; many intelligent environments exist in a single building and can intereact with users or other intelligent environments.

An intelligent environment uses available computational power to understand speech, recognize gestures, and comprehend facial expressions, so that it can interact with humans in a natural and intelligent way. For example, when a user says "start the presentation," an intelligent meeting room might close the drape, turn off the light, turn on the projector, and run the presentation software.

Intelligent environments will make our lives easier. However, with computers embedded everywhere from a coffee mug to a bathroom wall, security and privacy issues become more serious than ever; imagine a stranger who has access to an intelligent house and uses it to monitor and analyze the habitants from the bathroom to the bedroom.

1.1 Problems and motivations

Security and privacy are apparent and serious problems in intelligent environments. However, the research community views these problems as an afterthought. Moreover, many research articles touch on these subjects, but their focuses are usually on encryption, which is not enough to handle complex scenarios in intelligent environments. It seems that the problem of security and privacy in intelligent environment has not been studied and understood thoroughly.

1.2 Goals

The goals of this thesis are to deepen the understanding of these issues and present a conceptual model that solves security and privacy problems in intelligent environments consisting of multiples users in multiple locations. Our goals can be accomplished by trying to answer the following question: "Who has access to what kinds of resources, and under what conditions?"

1.3 Scopes

To realize those goals, we choose to limit the scope of this thesis using the following criteria.

1.3.1 Trusted environment

We will address the problem of security and privacy only in a trusted environment. A trusted environment means that every entity in the environment (e.g. humans, devices, spaces) adheres to the design requirements, such as notice and choice and consent, described in Chapter five of this thesis. Note that an entity can still lie to or have malicious intentions toward other entities; the solution and model presented in this thesis will still hold as long as those requirements are satisfied and maintained.

1.3.2 Civilian Solution

This thesis will focus on intelligent environments intended for civilian uses. Since our goals are to solve the problems of security and privacy while enabling useful and complex interactions, we believe that our solution might not be suitable for military or federal organizations, where security is of primary concern.

1.3.3 No indirect data deduction

Regardless of how well intelligent environments handle the issues of privacy, humans still lose privacy through other channels. For example, a user signs up for a service on a website by providing personal information. This personal information is often sold to third parties for marketing purposes. The information that an entity obtains from intelligent environments might be trivial. However, this information, when combined with information from other channels, might allow new kinds of information to be deduced. For example, John knows from Mary's friend that she likes to eat pepperoni pizza with orange juice. If he requests access to a video camera in a conference room and sees that there is a box of pepperoni pizza and a can of orange juice, he can deduce that Mary probably was in the conference room not long ago. This thesis will not cover this particular kind of case, because intelligent environments have no way of obtaining information from other channels.

1.4 Approach

The approach that we use is to show that the problem of deciding who has access to what resources can be broken down into one or any combination of three aspects: security, privacy, and quality of service (QoS). We present an access control mechanism for intelligent environments that incoporates the concept of security, privacy, and QoS using contextual information as an answer to the question in section 1.2.

1.5 Organization of this thesis

We begin in Chapter two by defining terminologies used in this thesis and covering background information necessary to understand the later part. Chapter three discusses scenarios that can happen in intelligent environments. Chapter four presents related works that have been done in access control mechanisms. Chapter five outlines requirements for access control in intelligent environments. Chapter six illustrates how access control, which address the issues of security and privacy, and provides a framework to address the issue of QoS, can be implemented in intelligent environments. Chapter six discusses the advantages and limitations of implementation. Finally, Chapter eight lists contributions that we have made to the field of intelligent environments.

Chapter 2

Background

This chapter will cover the background necessary for a better understanding of the materials in other chapters. Terminologies and the basic concept of security, privacy, and intelligent environments(IEs) will be explained.

2.1 Terminologies

Requester

A requester is a person who requests access to resources not belonged to her. For example, a visitor might request that a light be turned off at the room she is currently in. The visitor is considered the requester from the room's point of view.

Contextual Information

In this thesis, contextual information is the information that is not hard-coded into the system. Often, this information is dynamic, meaning that it can change from time to time. The system gets contextual information through many channels. Contextual information can be received from the sensors within intelligent environment or be given by another society.

Positive Acknowledgment

Positive acknowledgment means that an acknowledgment is required prior access to resource. For example, if John want to use Mary's telephone, Mary needs to explicitly grant access to John before he can use it.

Negative Acknowledgment

Negative acknowledgment assumes that access to resource is granted unless notified. For example, if John want to use Mary's telephone, he can tell Mary that he want to use the telephone and start using it right away, unless Mary states that John cannot use the telephone.

Quality of Service(QoS)

QoS is defined as a number that describes user's convenience. For example, if a user wants to print a file on a printer. The nearest printer would have higher QoS number than other printers, because it is more convenient for the user to pick up the document.

2.2 Security

According to Webster dictionary, security is defined as "measures taken to guard against espionage or sabotage, crime, attack, or escape." When applied to a computer system, security protects the resources by preventing access to an unauthorized entity. Security can be implemented at many levels ranging from low-level hardware (i.e., separate virtual memory access from different programs) to high-level software (i.e., serial number for software installation).

There are three important components in security: authentication, authorization, and secured communication. Perhaps an example can illustrate the function of each component. Let's say Ben is in California and he uses secure shell(ssh) to log in to his account at MIT with his username and password. Once logged in, he can manipulate

files in his own directory. However, If he tries to access a file in Alice's directory, the server at MIT will not allow him to access any file in Alice's directory. In this example, Ben authenticates himself by using his username and password, so that a server will know that it is communicating with Ben. When Ben tries to access files in Alice's directory, the server needs to check if Ben is authorized to access Alice's files or not. Finally, key stroke commands sent by Ben from the west coast to the east coast are encrypted to ensure secure communications, so that no attacker can impersonate or eavesdrop on information transferred between Ben and the server.

The level of security in any system is equal to the security of the weakest link in the system. As a result, authentication, authorization, and secure communication are of equal importance. This thesis will focus on access control mechanisms for intelligent environments, a core element in authorization.

2.3 Privacy

Marc Langheinrich divides privacy into four categories: territorial privacy, communication privacy, bodily privacy, and information privacy [11]. Our work focuses on protecting information privacy, defined as the ability to control one's own information, by incorporating the notion of privacy into access control mechanisms.

Private information can be further divided into two categories: static information and dynamic information. Static information is the information that does not change very often and does not require any deduction to understand the information. Credit card numbers, and social security numbers are examples of static information. Dynamic information is the information that changes often and requires some forms of analysis, so that it can be understood. Internet users' behavioral profiles created by monitoring their activities in the website represent an example of dynamic information.

2.4 Intelligent Environments

We define an intelligent environment to be the environment or space in which computers are embedded in the environment and they can interact with users with intelligence (i.e., react to users and understand users' speech). This section aims to give a brief introduction to issues in IEs that are relevant to this thesis. More details about some of these issues can be found in [5].

2.4.1 Software Agent

This section generally defines what a software agent is and why software agent architecture is useful for intelligent environments.

MIT's Intelligent Room project defines a software agent as "any software object with the ability to communicate by exposing functionalities to other agents running within the networks." [9]. An intelligent environment can consist of a collection of hardware and software components. As a result, it would be inefficient to implement a completely centralized system to control these components. Moreoever, a centralized system would suffer from a single source of failure problem. What an intelligent environment needs is a distributed system where each component would operate on its own and be able to communicate with others.

Since most software agent architectures are distributed systems with a small degree of centralization, they are good candidates for use in intelligent environments.

2.4.2 Resource Management

In a distributed system such as an intelligent environment, an agent needs to be able to locate another agent in order to communicate. Moreover, there is a need for arbitrations when two or more agents want to access the same agent that controls specific resources. The problems of resource discovery and arbitration are resource management problems. In this thesis, these problems are assumed to be solved by a software module called a resource manager (RM). More details of the resource manager can be found in [7].

2.4.3 People, Spaces, and Societies

Three types of interaction exist in intelligent spaces: interaction between a space and users, interaction between multiple spaces, and interactino between multiple users [9]. The first type of interaction occurs when one or more users are in the space. For example, a user can ask the space to display a file, or a space turns on the light when users enter. The second type of interaction can happen when one space requests a service from another space. For example, a server room might send a request to a control room to lower the temperature of the central air condition, when it detects that the temperature in the room is too high. The last type of interaction occurs when users interact with each other. Note that an interaction in intelligent environments can be a combination of these three types.

Since different entities (i.e., people and spaces) may have different goals, it is necessary for intelligent environments to have a framework for arbitrating resources between entities. This constraint prompts the needs to divide devices and software agents into groups so that each group can manage its own resources. To address this need, we introduce a concept of a society. A society is a collection of devices and software agents that act on behalf of an entity. Communications between societies are routed through each society's ambassador agent that acts as a proxy, so that resources and functionalities can be exposed selectively through the use of a resource manager and access control.

2.4.4 Access Control in an intelligent environment

The role of access control is to decide who has access to what kinds of resources and under what conditions. Based on this definition, there are three aspects of access control: security, privacy, and quality of service. In the security aspect, access control will focus on a requester and contextual information, such as location and activity. Access control might have the set of rules to determine whether the requester has the right to access a particular resource or not. In the privacy aspect, access control's focuses include a requester, a person whose privacy might be violated, and contextual

information. In the quality of service aspect, access control takes into account quality of service factor from each requester and allow resource access that results in the maximum utility.

In traditional computer systems, only security and privacy aspects are prevalent, and access control rules in these systems are rigid. However, in intelligent environments, access control rules do not have to be rigid. In some cases, it might be useful to violate the rule if the benefit for violating such a rule is higher not that of not violating the rule. For example, delivering an urgent message may be worth violating the no access rule for the messaging agent.

2.5 The distinction between security and privacy in this thesis

Our approach to security and privacy problems in intelligent environment is to sort them out and deal with each of the issues separately. Throughout the thesis, we discuss about an access control rule that focuses on security, but not on privacy. As a result, it would be less confusing to draw out the distinction between the meaning of security and privacy in an access control rule beforehand.

Security and privacy are often overlapped in the design of access control rules. When we talk about security, we often think of protecting access to resources or information. When we talk about protecting information privacy, we also think of protecting access to resources or information.

In this thesis, a rule that focuses on security will allow a requestor to access the resource, because he has the priviledge to use it. On the other hand, a rule that has a privacy component will allow the requester to access the resource only if doing so would not violate someone else's privacy. For example, a rule "any professor can access a video camera in a conference room." does not have any privacy component in it; allowing a professor to access this video camera during a conference might violate someone's privacy. On the other hand, a rule "any professor can access a video camera

in a conference room as long as no one is in the room" has a privacy component in it, because it takes into account the fact that the video camera should not be access if there is a person in the room.

Chapter 3

Scenarios in intelligent environments

The scenarios presented here range from the simplest one involving only one user and one space to the most complex one involving multiple users in multiple spaces and organizations. Each scenario is analyzed and problems associated with each scenario are presented. Key issues derived from these problems will influence the requirements and the design decisions for our access control described in Chapters five and six.

3.1 One user and one space

This scenario involves only one user and one space. We assume that only the user has access to the space. This scenario can be applied to a bedroom, a bathroom, a living room, or an office that only one person has access to it. In this scenario, authentication and QoS are relevant.

3.1.1 Security problems in this scenario

If there is only one person accessing the resource, an authorization component, which is the focus of this thesis, is irrelevant; all access can just be granted to this person, so access control needs not be complicated. However, the issue of authenti-

cation is important, because an impersonator can gain access to every resource once authenticated.

3.1.2 Privacy problems in this scenario

In this scenario, privacy problem can be bundled as a part of security problems. If no one else beside a user will have access to resources, information privacy is automatically protected.

3.1.3 QoS problems in this scenario

QoS problems do exist in this scenario, but these problems are handled by a resource manager. For example, a user can request a particular service instead of specifying devices or resources. Such a request might be "Show me this document." The resource manager needs to decide what kinds of resources to use (i.e. printer or projector) to satisfy the user's need with the highest uility.

3.2 Multiple users and one space

This scenario involves multiple users in one shared space. A setting can be a room in a house shared by roommates or an office space shared by two or more employees. In this scenario, we assume that access can be made only when a user is present in the room, and users trust each other to some extent.

3.2.1 Security problems in this scenario

One clear distinction between this scenario and the previous one is that resources can be shared. Resources existed in a space can belong to a specific society and the concept of private and public resources must be introduced. A public resource is a resource that can be used by any user, such as an agent that controls the lighting in the room. A private resource is a resource that belongs to a user, which can be shared with other users or not.

Two security problems arise in this scenario. First of all, who should regulate the access control. In the previous scenario, the room can regulate all the access control because there is only one user. In this scenario, private resources should be regulated by their owner, while public resources might be regulated by the space. By disseminating control based on the ownership, each entity can customize its own access control freely.

The second issue is overriding access control rules. In an emergency case, a user should be able to gain access to others' resources. For example, if there is a fire in a building, a user should be able to gain access to the server room, so that he can transport backup tapes to a safe place. Note that there is a potential for abusing this priviledge. A logging mechanism, explained in Chapter five, can be used to indirectly prevent these abuses.

3.2.2 Privacy problems in this scenario

When there are more than one people, access to resources can sometimes violate others' privacy. There are two cases in this scenario. The first case is when a user use his resources or public resources that can violate someone else's privacy. For example, a user can ask the space to record all the event when he is away. In the first case, a logging mechanism can be used, as a way to enforce a social norm. The second case is when a user request for a service that might allow others to violate his privacy. For example, Howie can ask his society to show him his bank account. However, a society might decide to show it on a big screen where other people can see it, thus violating Howie's privacy. To solve this kind of problem, a resource in a society must be classcified as privacy sensitive or not.

3.2.3 QoS problems in this scenario

When one or more persons request access to the same device, such as a projector, how can a resource manager make the decision. First of all, there are three societies involved. As a result, there are three resource managers. How and which resource

managers makes the decision can depend on resource type and violation.

First of all, the resource manager of a society that owns the resource has the right to make a decision. Let us assume that users request access to public resources belonging to a room. If there is no privacy problem involved, resource manager of each of the requester can submit its QoS to the room's resource manager. Then, the room's resource manager can give resource access to the society according to its maximum utility. On the other hand, if there is privacy problem involved, arbitrations between resource managers may be necessary. The arbitration process is discusses in Chapter seven.

3.3 Multiple users and multiple spaces in a single organization

This scenario involves multiple users and multiple spaces within a single organization. For example, John, who is in his office, might request access to a video camera in a playroom where Mary is currently eating her lunch. In this scenario, the restriction about user's presence has been lift; a user from outside a space can request access to resources inside the space. Removing this restriction makes security and privacy problems become more complicated. Contextual information becomes a key to solve these problems. Environment settings that match this scenario can be an office building, a school, a bank, a university etc.

3.3.1 Security problems in this scenario

In the previous scenario, we assume that there is trust between users, and users need to be in the space to make request. As a result, contextual information is not relavant as users can often arbitrate among themselves. However, in this scenario, the level of trust might be different based on a user's role. For example, we probably would not trust a janitor to allow him to access a bank's vault.

An organization comprises of users with various roles or functions. Each role has

different responsibility and privilege; each role may have different access to resources. In some situations, contextual information other than a role may be needed. For example, a user might not be able to access some resources from an unsecured location. As a result, an access control rule should have a framework to allow any users to easily integrate or extend existing rules with more contextual information as its attributes, such as location and activity.

Contextual information can be dynamic and subjective; a role of a user depends on what entity to which she makes a request. For example, Mary and John are close friends and they both work at a company called XYZ Corp. If Mary makes a request to use a resource belonging to the company, such as a printer in a meeting room, then her role, as perceived by the meeting room's society, will be that of an employee. However, if she makes a request to use a resource in John's society, her role, as perceived by John's society, might be a role of a friend.

3.3.2 Privacy problems in this scenario

By allowing a user to make a cross-space request, the privacy issue finally becomes more serious. For example, a request to access to a video camera in an occupied room can violate others' privacy. An access control rule can be created to protect privacy; a rule in the room might not allow any user to access the video camera if it is occupied. However, privacy is also subjective. Some person might not care about particular privacy violation, while others do. Furthermore, some users might be willing to trade off their privacy if the requester is willing to pay the cost. For example, a customer in a supermaket might allow his information to be gathered in exchange for a discount.

3.3.3 QoS problems in this scenario

In most organizations, there is a hierachy of roles, where the role at the top is the most important. For example, in a company, a CEO is probably more important to a company than a clerk. As a result, role or other contextual information might have impact on the decision making process of a resource manager.

3.4 Multiple users and multiple spaces in multiple organizations

In this scenario, we have multiple users, spaces, and organizations. This scenario is the most complicated scenario and it mimics real world situations. For example, an MIT student goes to a bank to withdraw her money. Her role with respect to the bank might be a role of a customer. Since her organization (MIT) differes from the bank, we assume that the level of trust may be low. Moreover, her location is currently at the bank. As a result, the bank might use resources to gather information about her, thus violating her privacy. Since each organization may have different interests, conflicts among them are likely to occur. These conflicts are the main problem in this scenario. When a user from one organization is in a location belonging to another organization, all sorts of violations can occur.

3.4.1 Security, privacy and QoS problems in this scenario

Each of these aspects suffer from the same problem: violations. When a user move into a different organization, security, privacy, and QoS might be violated. For example, some companies might not allow visitors to bring laptops or other electronic equipment inside their building. The visitors might be monitored by a video camera eventhough doing so might violate their privacy. They might also have limited rights and their QoSs might not be taken into account seriously by the resource manager of the space.

Moving into another organization might cause a user to lose security, privacy, and QoS. It might not be possible to prevent such violations because each organization has its own rules. However, users should be made aware in advance what rights will be taken away from them before entering any territory belonging to other organizations. Doing so would allow the users to weigh trade-offs and decide the best course of action.

3.5 Key issues for access control in intelligent environments

Here are access control issues categorized by the three aspects:

Security:

- Each society should havee its own access control.
- Access control should allow contextual information to be integrated in its rules.
- A logging mechanism is needed to prevent abuses.

Privacy:

- Privacy is subjective.
- Resource must be classified if it is privacy sensitive or not
- Advance notification is necessary when a user move into a different territory.

QoS:

- Contextual information and privacy must be taken into account when computing QoS.

Chapter 4

Related Works

In this chapter, we will first present a brief survay of research projects related to intelligent environments and how each of them addresses the security and privacy issues. The focus of the discussion will be on access control. Then, various access control mechanisms(ACMs) currently used in computer systems or intelligent environments will be explained. Their weaknesses, when they are used or if they are to be used in an intelligent environment, will be elaborated.

4.1 Current research projects in intelligent environments

Research in intelligent environments has been done in both academic institutions and commercial cooporation. We will focus on security and privacy aspects of intelligent environments when discussing each research project. Research projects in academic instituion includes the MIT's Intelligent Room project [5], the MIT's mobile network device project [3], the Georgia Tech's Aware Home project [13], the University of Illinois's GAIA project [14]. Research project in commercial organization includes Microsoft's Easy Living project [15].

Currently, MIT's Intelligent Room project does not have the notion of security or privacy in the room. As a result, this thesis aims to analyze access control prob-

lems in intelligent environment and lay the framework for future implementation of access control. MIT's mobile network device project focuses on mobile device. It uses the most primitive form of access control called access control list(ACLs). Georgia Tech's Aware Home project have Generalized Role Based Access Control(GRBAC). GRBAC can be considered a variant of Context Based Access Control(CBAC), which will be explained in section 4.2.3. The Univeristy of Illinois's GAIA project focuses on authentication that aims to protect only user's location information. The Microsoft's Easy Living project does not give any specification of how exactly security and privacy issues will be handled. However, [15] suggests that some kinds of resource classification (i.e., privacy sensitive resource) will be used to protect user's privacy.

4.2 Access Control Mechanisms

From the previous chapter, access control issues in intelligent environments prompt new requirements for access control mechanisms to be used in intelligent environments. Recall that the role of access control mechanisms is to decide who has access to what kinds of resources and under what conditions. In an intelligent environment, an access control mechanism must provide a framework to address the issues of security, privacy, and QoS. In order to address these issues sufficiently in intelligent environments, contextual information must be used in an access control mechanism. Unfortunately, none of the currently available access control mechanisms can address all three issues and deal with the variety of complex situations that can take place in an intelligent environment.

4.2.1 Access Control Lists (ACLs)

ACLs is the most primitive form of ACMs and the most widely used ACM. Essentially, ACL is a mapping between users and resources that they are allowed to use. Unix file system is one of many systems that uses ACLs. In Unix, each file has an owner, the one who creates the file. The owner can decide to give access to other users. Once given access, a user can continue to access the file until access right is

revoked.

ACLs are widely used because they simple, easy to implement, and perfectly adequate for most applications. However, the simplicity of this approach comes at a price. ACLs has two apparent weaknesses: its size and its lack of contexts. Since ACLs uses no hierachy, the size of access rules scales with respect to the multiplication between the number of users and resources. Secondly, ACLs does not use any contexts. As a result, its rule is not as expressive enough to be used in intelligent environments. For example, a rule "if it is raining, all use of equipment sensitive to lightning is prohibited" cannot be implemented using ACLs. In this example, two contexts, weather and equipment properties, must be integrated into a rule. An example of an intelligent environment that uses ACLs is [3]. ACLs might be adequate for scenarios discussed in sections 3.1 and 3.2, but it will not be able to perform adequately if used in scenarios discussed in sections 3.3 and 3.4.

4.2.2 Role Based Access Control (RBAC)

RBAC [6] is an access control mechanism centered on a role of a person. It is an improved version of ACLs. Instead of having a mapping between users and resources, RBAC uses hierachy by introducing a new attribute called a "role." In RBAC, a user can have one or more roles. Based on role(s), the user is allowed to perform specific operations on an object. For example, in a bank, a user can take on a role of an employee, a secretary, a teller, an auditor, and a manager. A teller can modify a bank account, while an auditor can view a bank account but cannot modify it.

RBAC also allows inheritance so that a new role can be built upon an existing role. For example, a role of an employee has access to the front door of the bank. The roles of a secretary, a teller, and a manager can inherit from the role of an employee. Role inheritance reduces the amount of rules that need to be implemented.

RBAC has been used in civilian organization, such as banks and hospitals [4, 2], because it reflects well the real life organizational structure. However, RBAC does not use any contextual information other than a user's role. As a result, the example rule mentioned in section 4.2.1 also cannot be implemented in RBAC. RBAC will not suit

for scenarios discussed in sections 3.3 and 3.4 as it does not allow other contextual information in its rule.

4.2.3 Context Based Access Control (CBAC)

CBAC [16], as its name implies, uses contextual information as its attributes. It augments RBAC by allowing other contextual attributes in its rules. In addition to a role attribute, time, location, and resource types can be incorporated into rules. While the original focus of CBAC is in medical database domain, it can be modified to include other contextual information available in intelligent environments. The example in section 4.2.1 can be implemented using an augmented CBAC to include weather as one of its attributes.

Eventhough CBAC and its variants allows complex rules that work well in database domain, they do not have a nice framework to address privacy problems in intelligent environments. Privacy issue is subjective and privacy preference for a user can change from time to time. As a result, while they allow contexts to be integrated into it rules, it would be difficult to use them as a sole access control in intelligent environment, especially for the scenario described in section 3.4.

4.3 Conclusion

Interactions between people and intelligent environment can be complex. As a result, an access control mechanism must provide a framework to allow a complex rule that reflects real situations within intelligent environment. Moreover, an access control mechanism must provide a framework to deal with conflict resolution between multiple users. Finally, there are also issues of privacy and quality of service. Unfortunately, none of current ACMs seems can handle all the issues mentioned. In the next chapter, design requirements for an access control mechanism in intelligent environment will be outlined.

Chapter 5

Design Requirements for Access Control in intelligent environments

In previous chapters, we have shown that there are three aspects of access control mechanisms in an intelligent environment: security, privacy, and quality of service. This chapter will elaborate on the design requirements for each of the aspects.

5.1 Design requirements for Security

Various papers describe how to design security in a computer system. However, very few of them describe how to design security in intelligent environments. This section aims to cover some of the main security concepts necessary for intelligent environments.

5.1.1 Contextual information

As described in various scenarios in Chapter three, contextual information is a required ingredient for access control in intelligent environment. Since each environment might require different contextual information (i.e., a school versus a bank), access control must allow new contextual information to be integrated easily.

5.1.2 Logging

Request for access to resources in some situations, such as during an emergency, should be logged. Note that logging does not directly make a system more secured; it exists so that if a security breach happens, logged information will allow authorities to pinpoint a culprit or weaknesses in the system. Moreover, logging indirectly enforces social norm; if people know that their access to resources are logged, they will unlikely abuse their usages. In some other cases, security can be violated if the QoS is high. Logging will also allow a user to audit the past decision to see if it is appropriate.

5.1.3 Active Security

In this thesis, there are two types of security: passive security and active security. Passive security protects resources by granting or denying resources only when a user requests resources. On the other hand, active security decides who has access to what kind of resource based on contextual information even though a user does not request access for resources. Active security can also involve retracting access to resources from a user.

A user is dynamic in a sense that he can change his behavior or role in a matter of second. As a result, a component in security should also be able to adapt to a user's dynamic behaviors, which is the purpose of active security. For example, if robbers enter the bank, access to bank's vault will be disabled, so that no one can open it. Unfortunately, few security systems nowdays provide or implement active security.

5.2 Design requirements for Privacy

This section introduces design requirements that are necessary to protect user's privacy in intelligent environments. An excellent overview of overall principles related to privacy issues can be found in [11]. This chapter covers two of the principles mentioned in the paper (e.g. Notice and Choice and Consent).

5.2.1 Notice

The principle of notice requires that advance notification be given to any user that will be affected, either directly or indirectly, by any information-gathering practice. In other words, if a user wishes to use his resources or access another society's resources to gather information about other users, his society fist has to notify them about the kind of information it wants to gather and what it will use that information for.

For example, if a video camera in an airport is used to monitor passengers in the terminal, these people should be made aware of what kind of information this particular video camera gathers, who will have access to this information, and how this information will be used. Note that this requirement is already violated at Logan Internation Airport where every passengers are identified by face recognition technology without advance notification. However, we feel that this requirement is important and should be implemented.

Advance notification is required so that an affected person becomes aware of her information being gathered and she can decide to accept the practice or not. A generally acceptable example might be a video camera that runs face recognition software to detect terrorists and criminals at the airport. In this case, only face information is gathered and faces that match one of the targeted criminals will alert security personnel.An generally unacceptable example might be a video camera that runs face recognition software to identify people and notify their creditors of their locations. Even a simple video camera can be used in either acceptable or unacceptable ways. As a result, the principle of notice is needed to keep affected parties aware and informed about the use of their information.

5.2.2 Choice and Consent

The principle of choice and consent requires that a consent from an affected party be obtained prior to any information-gathering. A consent can be either explicit or implicit. In intelligent environments, a consent need not be explicit; it would be very inefficient or impossible for a user to give explicit consent to one hundred sensors

around the area as he walks by. The solution for this issue is to create a privacy profile for each entity. More details about the privacy profile will be covered in Chapter 5.

In some other cases, a consent can be indirectly assumed by negative acknowledgment, if prior notice is given. For example, if a passenger is notified that his identity information will be gathered once he enters the airport terminal, his decision to enter the terminal indirectly implies his consent. Otherwise, he would have chosen not to enter the airport terminal.

The above example also demonstrates that sometimes there will be a trade off between QoS and privacy. That passenger might not be happy about his information being gathered in the airport, but he is willing to trade off his privacy so that he can travel to his destination by a plane. The role of this principle and the notice principle are simply to keep him informed, so that he can make a decision (e.g. to enter the airport or to walk away) that results in his best interest.

5.2.3 Resource Classification

As described in section 3.2, resources within a society must be specified if it is privacy sensitive or not, so that the resource manager will be able to select an appropriate device when displaying them. The resource type can be thought as one of contextual information that will be integrated into access control rules

5.3 Quality of Service

In intelligent environments, the concept of QoS has been used by resource managers to decide whether and what resources to give to users so that the overall utility is the highest [7]. For example, if two users request an intelligent room to display texts, the room would decide to give a larger monitor to the user who has more text to display. Since an access control mechanism has to cooporate with a resource manager, it should have the framework to support the concept of QoS. For example, a person with a role of the CEO should have higher QoS to the resource manager in a space compared to that of a person with a role of a janitor.

In the next chapter, I will explain how the concept of QoS can be fused with design requirements of security and privacy to provide access control in intelligent environments that is dynamic, flexible, and easy to conceptualize.

Chapter 6

Implementation

This chapter covers details of the implementation of the Intelligent Access Control (IAC). It is organized into three sections. The first section outlines the characteristics of IAC. The second section states necessary assumptions required for the IAC to be implemented and work correctly. Design decisions are covered in the third section. Finally, schematics and implementation details are elaborated in the last section.

6.1 Characteristics

Since interactions within an intelligent environment can be complex and dynamic, an access control for intelligent environments needs to keep up with these complexities. For example, a user should be able to tell his intelligent room, "I do not want anyone to reach me in the next two hours." In this example, the resource is the user and the access control needs to prohibit anyone from accessing this resource for the next two hours. This section describes four characteristics that an access control mechanism needs in intelligent environments: expressiveness, easy extension, adaptation, and a user centric framework.

6.1.1 Expressiveness

Expressiveness in access control means that a rule is complex enough to be used in a real situation in intelligent environment. A mapping between users and resources is not expressive. A rule in CBAC which allowes more contextual information such as time, role, and location, is considered expressive. The goal of IAC is to stretch expressiveness capabilities so that an IAC rule can be implemented for everyday life scenarios using any attributes, such as weather, mood, and activity.

6.1.2 Easy extension

Intelligent environments can be any place, such as a high school, a bank, and an airport. Each place might have the need to use different kind of attributes. As a result, IAC should allow users to easily extend IAC with more attributes as they see fit.

6.1.3 A User centric framework

As mentioned in the last few chapters, privacy and sometimes utility are subjective; there is no way to create a set of rules that will satisfy every user. As a result, an access control mechanism must have some kind of structure that is flexible for users to be able to define their own privacy profiles.

Note that the framework to support subjective issues is needed but it does not mean that privacy profiles will be honored. For example, a user might not want his identity exposed by face recognition at the airport, but it will not be possible for him not to be identified if he wants to use the airport.

6.2 Assumptions

Assumptions about properties of an intelligent environment and its infrastructure are made so that IAC's design can be built upon it.

Trusted environment

In this thesis, a trusted environment is an environment where every entity in the environment adheres to notice and choice and consent principles. IAC is designed to operate in this trusted environment. IAC can be used in an environment where security is the top priority, but it will be less efficient as most features will not be used. Instead, the focus of IAC is to make trade offs between security, privacy, and utility in ways that benefit users the most.

Authentication and secured communication

An intelligent environment shoud have a reliable way to authenticate users and secured communication channels between users and itself, so that no other entities can impersonate users or requests.

6.3 Design overview

Based on design requirements and characteristics, various design decisions must be made. The first issue is to decide where in the system IAC should be implemented. The next question is whether to build IAC from scratch, reuse, or augment existing access control mechanisms. The third issue is how the privacy profile, which is subjective to each person, should be implemented.

6.3.1 One access control for each entity

Intelligent environments consists of devices, software, spaces, and users. In Chapter two, we outlines how these entities can be grouped into separate societies. Each society represents either a space or a person; interactions between users and intelligent environments are basically interactions between societies. Due to the nature of the interaction discussed in Chapter three, each society should have its own access control as well as resource manager, so that it can selectively expose its resources and functionalities to other societies.

Within each society, access control should be implemented as an advisor to a resource manager; it advises the resource manager whether or not a set of requested resources should be given to a requestor or not.

The reason to implement access control as a part of a resource manager is because access control should be put at places where all requests will go through and a resource manager is one such place. In some systems where a resource manager does not exist, access control can be implemented in resource discovery modules.

6.3.2 Reusing available ACM

Chapter four establishs that current access control mechanisms are not suitable for an intelligent environment, because, they lack some frameworks to address relevant problems in an intelligent environment, such as privacy and QoS. However, more advanced access control mechanisms like Context Based Access Control can be incorporated as parts of IAC. Context Based Access Control can act as one of many modules in IAC designed to address issues of security, privacy, and quality of service. Specifically, it will be used to address the security aspect in IAC.

6.3.3 The Privacy Profile

The role of the privacy profile is to check if a user's privacy is violated and negotiate or notify the user based on circumstances. Privacy preferences are subjective and dynamic (i.e., can change over time) to each user. Moreover, attributes in preferences can be more complicated than attributes associated with the security part of access control. For example, a user might not care if a video camera is simply used to identify her face, but the same user might care if the video camera is used to record her shape.

Access to the same resource can violate privacy depending on what kind of information is gathered. However, the kinds of information that can be gathered from some resource, such as a video camera, are many. Quantifying this kind of information into various attributes can be troublesome.

Based on the dynamic and complicated nature of the privacy profile, we decide to use multiple weight-directed graphs as the data structure, because weight-directed graphs are easy to implement and versatile. Privacy profile will also output QoS that a resource manager can take into account when computing the utility. More details about privacy profile will be covered in the next section.

6.4 Schematics and Implementation details

This section outlines how IAC can be implemented in an intelligent environment using the requirements and assumptions discussed so far. It is organized into three sections. The first section shows the overall schematics, elaborates on minor modules, and discusses how the information flows among different societies. The second section discusses how to implement passive security and active security in IAC. The third section outlines the implementation of the privacy profile using weighted direct graphs.

6.4.1 Schematics and Information Flow

There are two phases when one society requests access to resources from another society: the request phase and the arbitration phase. In the request phase, a request is sent from a society that needs a resource to a society that has the required resource. There can be three outcomes: the request is granted, the request is denied, or the counter-proposal is presented. Figure 6-1 illustrates the request phase involving three societies. Three societies in this figure are societies Pipe, E21, and K. Societies Pipe and K represent human, while society E21 represents an intelligent space. The fact that society K is within society E21 means that a user K is in the intelligent space E21. Each society has a resource manager(RM), an access control (AC), and logging module. This particular example is presented because it can be easily applied to the request involving two or more societies. Figure 6-2 depicts IAC, which consists of an active security module, a privacy profile, and a passive security module.

In some cases, arbitration might be required. For example, K might not care

if somebody can view him eat his lunch as long as he knows where that person is. Figure 6-3 shows the arbitration processes, which continue from the request phase.

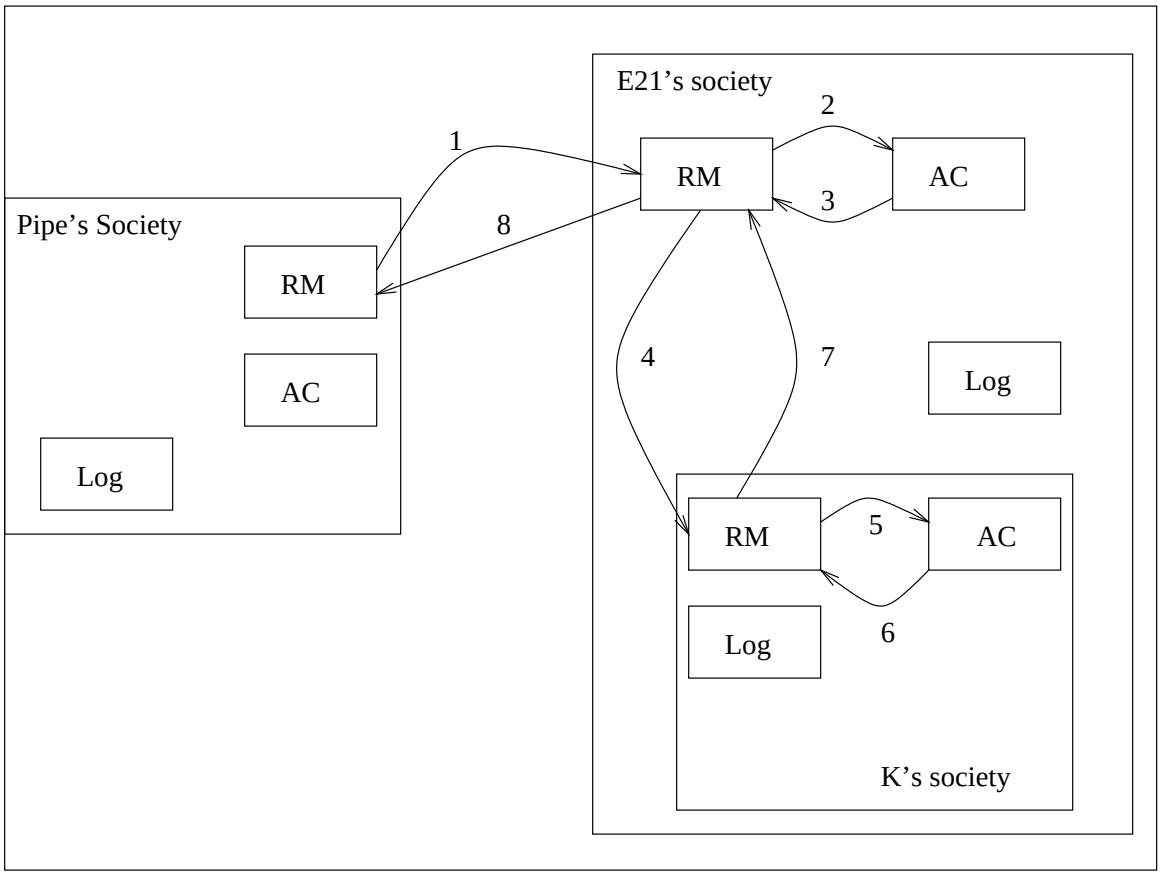

Figure 6-1: The conceptual flow diagram for multi-society communications in the request phase. A real world scenario that creates the kind of information can be the following: K is eating in the intelligent room E21. In another room, Pipe ask his society to request access to a video camera in room E21 (1), because he forgot his book somewhere and wants to check whether the book is in the room E21 or not. After society E21 receives the request, E21 checks with its access control to see if Pipe has access to the device or not (2). E21's access control finds out that Pipe has access to the video camera in the room (3). However, E21 society notices that K is in the room. As a result, allowing Pipe to access the video camera might violate K's privacy. Therefore, E21's society asks K's society if it is permissible to allow Pipe to use the video camera in E21 (4). K's resource manager checks with K's access control(5). In this case, K's society does not own the video camera. As a result, only K's privacy profile plays a role in the output from K's access control. K's privacy profile recognizes that Pipe is K's friend. As a result, it tells the resource manager that allowing Pipe to access the video camera will not make K loses any utility (6). K's resource manager checks the utility input from the access control and notifies E21's resource manager that it is permissible for Pipe to access the video camera (7). E21's resource manager then supplies Pipe's society with the stub that allows Pipe's society to control the video camera (8).

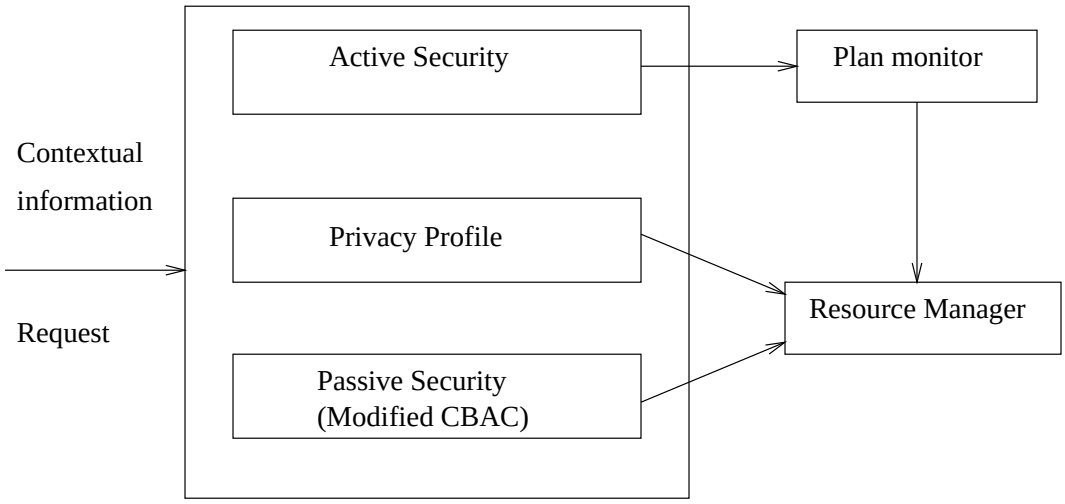

Figure 6-2: an IAC module receive two inputs: contextual information and a request. The output from active security will be sent to a plan monitor, while the outputs from a privacy profile and passive security will be sent to a resource manager.

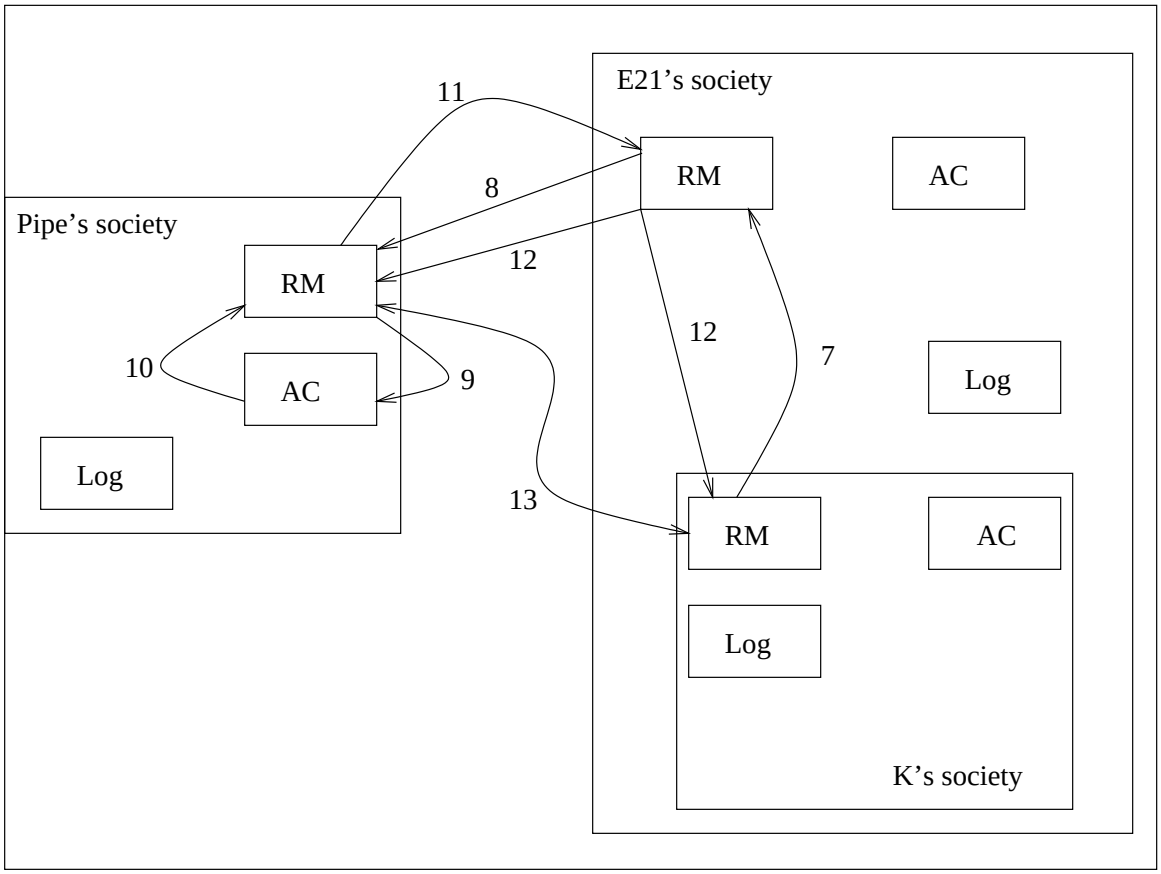

Figure 6-3: The conceptual flow diagram for multi-society communications in the arbitration phase, which continues from the request phase. Instead of telling E21's resource manager that it is permissible to let Pipe's society access the video camera, it tells E21's resource manager that it requires Pipe's society to divulge Pipe's current location (7). E21's resource manager then notifies Pipe's society of the request (8). Pipe's society checks with its access control to see if it is possible to divulge Pipe's information (9). Pipe's access control sent the result to the resource manager (10) so that it can analyze the utility. Pipe's resource manager decides that it is acceptable to divulge Pipe's location and notifies E21's resource manager(11). Then E21's resource manager concludes the arbitration, gives Pipe's society access to the video camera, and notifies K's society that Pipe's society accepts the request(12). Finally, Pipe's resource manager contacts K's resource manager and gives K's society a stub for Pipe's location agent (13).

Logging

After a transaction, the result can be stored in the Log. This section will answer two questions: who should keep the Log and what should be in the Log.

Any society involved in a transaction should keep a log. However, a society that controls shared device, such as the E21 society, should only keep a log when a request made during an emergency situation, so that requester's privacy is protected. A society that requests resources from other societies should keep a log of transactions, so that it knows what kinds of request are granted. A society that receives requests should also keep a log, so that if its resources are abused, it can find culprits. A society affected by a request, such as that in the example in 6.4.1, should also keep a log, so that the effects from such a request can be analyzed to help future discussions.

A log should contain a timestamp, a request, and an outcome of a request. A request is composed of a requester's society name, a list of resources, and the name of a society that receives a request. An outcome can contain the result (yes/no) or arbitration agreements.

Proxy and Inter-Society resource discovery

Not mentioned in the information flows are Proxy and Inter-Society resource discovery. First of all, it would not be wise to let other societies gain direct access to the resource manager. For example, allowing every society in the world to know that the society John-living-room has plasma display as one of its resources may tempt someone to steal it. As a result, there is a need for a proxy that acts as a society's interface to the outside world. In this thesis, an Ambassador agent acts as a proxy. It receives requests from another society and sends them to appropriate modules in its society. All the requests from its society to other societies will rely on the Ambassor agent to send them out. The reason for having an Ambassador agent is to allow any society to hide its resources and selectively expose only some resources to the outside world.

Secondly, an ambassador agent needs to be able to find other ambassador agents.

To allow such a communication, an Ambassador agent needs an interface to a wide scale resource discovery modules, such as the Intelligent Naming System [1], One-dot-World project [12], or Hyperglue [8].

6.4.2 Security implementation

Figure 6-2 shows an IAC module in a society, in which there are two components in IAC that are used to address security issues: passive security and active security. The modified CBAC is used for passive security, while REBA is used for active security. Active security has the highest priority in IAC, because it handles emergency situations. The plan monitor keeps track of a plan that a society needs to do to satisfy a current goal.

The purpose of security components is to address only security issues. CBAC and REBA should not have any rules that try to cover privacy and conflict resolution issues, which are privacy profiles and resource manager problems, respectively. The idea of trying to cover security and privacy in a single access control might be tempting, as it has been attempted in RBAC and CBAC. However, as explained in previous chapters, it would be best to separate security and privacy problems.

Passive security take a request for resource access (i.e. a requester's society name and a list of resources) and contextual information, and output either grant or deny to the resource manager. However, the resource manager also must take into account of the output from privacy profile. It is possible for the resource manager to bypass passive security if the QoS is high enough.

Active security analyzes the request to see if it is consistent with the current context according to the plan monitor. If not, it notifies the plan monitor. The plan monitor then decides the best course of action and advise the resource manager what to do.

Passive Security

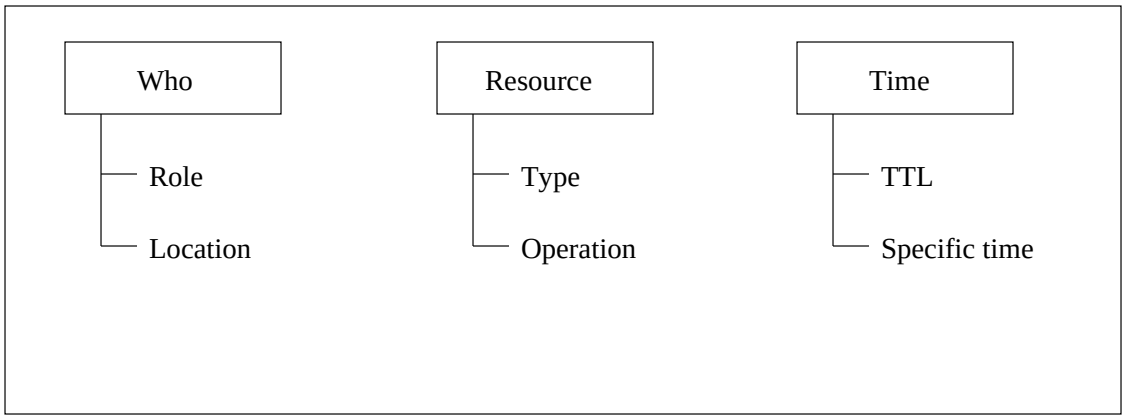

Figure 6-4: One possible configuration for the modified CBAC. This particular configuration consists of six attributes sorted into three categories. More categories or attributes can be added to make a rule more expressive. An example rule for this figure might be *if (role=employee) and (location=inside) and (type = printer) and (operation = print) and (specific time=Monday) then the request is granted*

Using concepts described in [16], one can implemented CBAC and augment it with more attributes. Figure 6-4 shows one possible attribute configuration for the modified CBAC. One way to implement modified CBAC is to use multiple hashtables where the value of the first hashtable becomes the key of the second hashtable. More attributes, such as weather and activity, can be added as needed. This configuration shows only attributes that can be divided into three big categories: *who, resource,* and *time*. The who category contains attributes related to a requester: role and location. The role attribute is derived from the role attribute in RBAC, while the location attribute is the location of the requester. Resource category contains attributes related to resource such as resource type and operation. The resource-type attribute will describe what kind of resource it is (i.e., hardware, software, database, and content). The operation attribute specifies allowed operations for the resource. Finally, time category contains a time attribute which can be specified as time-to-live (TTL) or specific time and date.

How can one design rules in CBAC that only address the security aspect of access control? A simple clarification about the root of privacy and quality of service problems might help. The reason that privacy problems exists is that there is more than one person in the world; if there is only one person in the world, then it is not possible

for that person to violate others' privacy. The QoS problem exists because resources are limited and some resources might suit a particular task better. By ignoring these constrainsts, rules that focus on security aspect can be created.

With these assumptions in mind, rules become simpler to implement and easier to conceptualize, argue, and reason. For example, a user who is outside a meeting room should not be able to control the light in the room, if there are people in the room. To implement this rule for the meeting room in normal CBAC, four attributes must be used: user, user's location, resource type, and number of people in the room. The rule might be *if (role=rule) and (location=outside) and (resource = light) and (people != 0) then the request is denied* However, a rule in the modified CBAC would use only two attributes: user and resource; *if (role=user) and (resource=light) then the request is granted* The reason for using two attributes is that this is simply a QoS problem. If a user can use the light when she is outside the room, it an reduce QoS of any person in the room. The resource manager will deal with QoS problem directly and leave security components to focus only on the security aspect; the security rule might allow a user to control the light, but the resource manager can decide not to let the user control it because it violate others' QoS. The result is that a rule becomes less complex, which means less potential for errors.

Even if a rule focused on security aspect can often be created by using only a mapping between user and resources (ACLs), contextual information can be required in some cases. The rule in section 4.2.1 that requires the weather attribute is one example.

Active Security

As mentioned in Chapter five, active security decides whether or not to grant users access to resources based on current context with respect to the current plan of a society. Active security is designed to be used in emergency situations where normal rules regarding security, privacy, and quality of service might not apply.

The active security module can be implemented using the Reactive Behaviroal system (REBA) [10]. REBA is a system that reacts to users based on contextual

information from sensors within an intelligent environment. REBA comprises a set of behaviors or modes that can be specified to have a specific topological ordering; one behavior can override only certain behaviors. For example, a vacant behavior can be overridden only by an occupiecd behavior.

A behavior is active when certain conditions are met. For example, an occupied behavior is triggered when a user enters a room through a door. Each behavior can trigger certain accesses or commands to resources. However, when a behavior is overridden by another behavior, certain accesses or commands to resources can be masked with new accesses or commands. For example, the occupied behavior can trigger a command to a light manager to turn on all the lights in the room and a command to a fan manager agent to turn on all the fans. However, a presentation behavior, which becomes active when there is a meeting in the room, can override the command to open all the lights by the occupied behavior while still leaving the command to the fan manager agent active.

Using REBA, one can implement active serucity by creating a set of behaviors for a specific environment. When REBA's behavior changes from one to another, instead of triggering access or commands to a resource manager itself, we make it notifies the plan monitor to check if the new behavior is consistent with the current plan. If not, the plan monitor can advise the resource manager to grant or repeal certain resources. More details about REBA implementations can be found in [10].

6.4.3 Privacy profile implementation

Since privacy is subjective, each society that represents a human user should have his privacy profile within his own society; privacy preferences are also considered private information. The role of a privacy profile is to determine whether other societies violate a user's privacy or not. This section outlines specifications for a privacy profile, such as input/output parameters, data structure, and a decision making algorithm.

Input/Output

Privacy is violated because of information that can be derived from accessing resources. For example, one can use a video camera to gather information from other people, such as their behaviors, activities, and friends' identities. On the other hand, just because someone has access to a video camera does not mean that such information will be gathered; a video camera in an airport might be used only for facial recognition.

Based on the above argument, there are three important factors to consider. The first factor is a requester. Specifically, can the requester be trusted? The second factor is what kinds of information will be gathered and how the information will be used. The third factor is the potential information and potential uses of it that can be gathered from a requested device.

We propose that there should be three kinds of input for a privacy prfile. The primary input is the requester identity. If the requester can be trusted, the second input should be information he intends to gather and his intended usages of that information. Otherwise, the second input should be potential information and its potential uses. The last input should be other contextual information, such as location and activity. This contextual information will help a privacy profile determine if the request is appropriate and in the user's best interest.

The output from the privacy profile will be used by a resource manager to weigh the decision and to conduct an arbitration if necessary. Note that in some circumstances, privacy can be violated if the quality of service is high enough or the requester is willing to pay a price as described in section 3.3.2. As a result, the finer granularity output than yes/no is needed. A range of integer can be used as one of the outputs where higher results mean a user would prefer not to have his privacy violated. We called this integer output QoS. The other output can be viewed as a cost or demand for violating the user's privacy. This output can be resources belonging to the requester, such as money.

Data Structure

The data structure in a privacy profile will be multiple weighted direct graphs, where each graph will have a starting node l of type location and an ending node t; each location will correspond to a specific graph. If there is a path from l to t, then an output is available for a resource manager to make a decision or conduct an arbitration. The output consists of a flow from l to t. The flow will contain nodes and arcs with associated arc costs.

The reason for using location as a starting point in a graph is for the purposes of simplification. Location often defines a set of possible activities. For example, a person would probably not sleep in a restroom. Within each activity, a role for a requester with respect to a user can be defined. Note that the role of a person is different with respect to different people as desribed in section 3.3

Using graphs is advantageous. First of all, the privacy problem can be modeled as a combination of a shortest path and a maximum cost flow problem. Secondly, an arc cost can be used to signify how much a user will tolerate the violation of privacy. Since graphs for a privacy profile will not have any cycle, computing results will not be difficult and time consuming.

Types of a node in a graph can be unbounded, because each person has different preferences. However, for simplicity, every node in this thesis will be one of the following types: location, activity, role, user's information, and cost. Location and activity nodes are the user's, while the role node is the requester's. The user's information node contains information about the user. Note that location and activity can be considered the user's information. However, different designations will be used to prevent a cycle in the graph. A cost node specifies the cost that a requester has to pay to violate the user's privacy. Figure 6-5 shows an example.

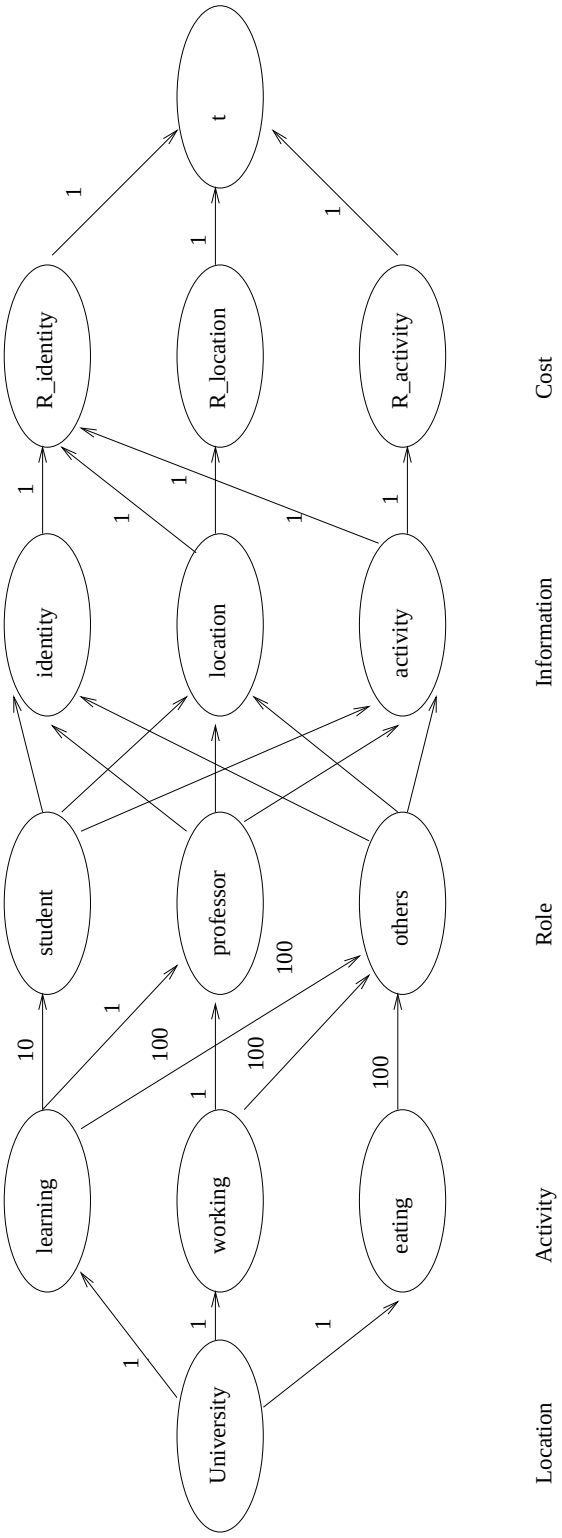

Figure 6-5: An example graph

Decision Algorithm

The output decision is made by using a modified shortest path algorithm, a maximum flow algorithm, and contexual information. Contextual information should allow us to construct a path from a location node to an activity node and from an activity to a role node. However, since a requester can have more than one role, a role with smaller arc cost will be selected. Then, a maximum cost flow algorithm can be used to compute the flow that includes all information nodes (plus their associated cost nodes) that a requester wants. If there is a path from the location node to the end node t, then it means that the output can be computed.

Let's use Figure 6-5 as the example. If a user is eating, then there is only on path from the lcoation node to the role node. The path university, eating, others will have a total arc cost of 101. Let's assume that a requester wants to access a device that divulge the user's identity and location. The algorithm will assign edges *(other, identity)* and *(others, location)* with capacity one, and the edge other,activity with capacity zero. The next step is to run a maximum flow algorithm starting from node others to node t by assuming all other edges not mentioned will have an infinite capacity. Figure 6-6 shows the graph starting with the node others. Note that c/f/u for each edge means cost/flow/capacity. However, the edge with an infinite capacity will only have c/f (cost/flow).

After running the max flow algorithm, the flow is shown in Figure 6-7. Next, we can sum the cost of edges that belong to the flow which equals to 304. As a result, the total arc cost QoS will be 405, and the cost nodes will be $R_{identity}$ and $R_{location}$. These outputs from the privacy profile will be the inputs into a resource manager.

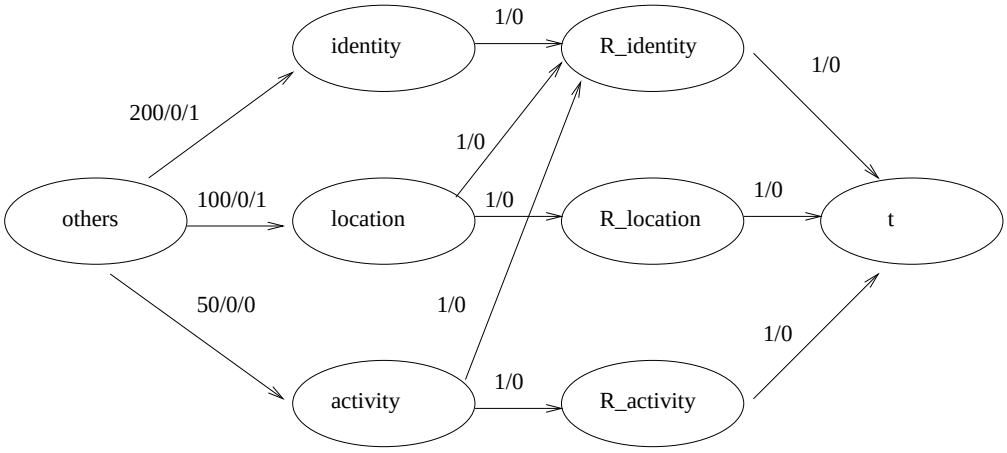

Figure 6-6: The input into the maximum flow algorithm. In this example, accessing the resource will violate user's privacy by divulging his identity and location. As a result, edges *(others,identity)* and *(others,location)* are assigned with capacity of one

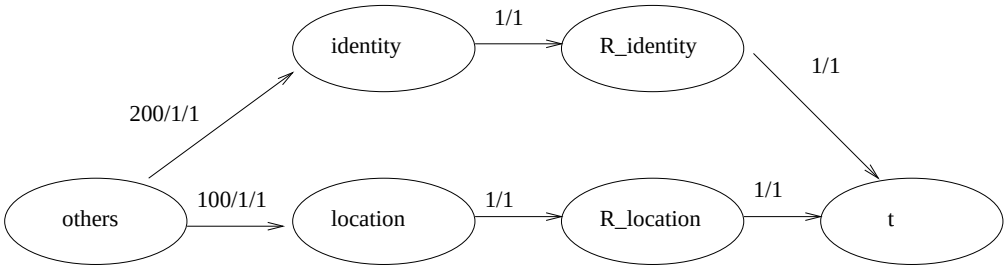

Figure 6-7: The output from the maximum flow algorithm, which are the flow and the sum of all the arc cost (QoS)

Chapter 7

Discussion

7.1 Advantages

7.1.1 Expressiveness

Using context in IAC allows a rule that is expressive and can reflect real situations in intelligent environments. In fact, it would probably be impossible to implement usable access control in a multi-user multi-space intelligent environment where resources are shared, because the situation is so dynamic (i.e., role changes or users move in/out of the room) that access control mechanisms, such as ACLs, and RBAC, will not work.

7.1.2 Divide, Conquer, and Combine

IAC breaks the problem of deciding access to resources into three sub-problems (security, privacy, and quality of service). Each sub-problem is mostly handled by a specialized module, except for a privacy profile that also has a QoS component, and results from all modules are combined to formulate the final answer. Using this approach reduces the complexities of the implementation; instead of having a rule that deals with security, privacy, and quality of service all at once, we can have security modules handle security, a privacy profile handle privacy problems, and a resource manager handle the quality of service problem.

7.2 Limitations

7.2.1 Strong dependence on contextual information

Contextual information allows IAC to be sophisticated, intelligent, and powerful. It would be difficult to implement access control in intelligent environments that is expressive without contexts. Hence, IAC strongly depends on contextual information to make a decision or recommendation to a resource manager.

Contextual information can be acquired through various types of sensors or applications. For example, temperature context can be acquired from a software agent that reads the output from a thermometer, while visual context such as user's activities can be acquired from an agent that interfaces with vision processing applications.

More attributes in IAC means more contextual information is needed. If sensors or applications fail, then it is possible that IAC might not be able to reach a decision. Worse, if contextual information is wrong, IAC can make a wrong decision. Our current suggestion is that contextual information should be selected and received from trusted and reliable agents.

7.2.2 Trade-off between expressiveness and complexity

More attributes allow rules in IAC to be expressive. However, they also make them more complex. As a result, it will be difficult for a human to audit all rules in access control to ensure that it is correct and error free. We have thought of using some kind of machine learning algorithms to keep track of rules and their evolutions.

7.2.3 No way to enforce the separation of security and privacy in the implementation

Although we argue that a rule focused on security should be implemented in CBAC and a rule focused on privacy should be implemented in a privacy profile, there is no way to enforce this kind of separation; a rule in CBAC might be implemented to protect a user's privacy even though it should not. For example, Pipe might not want

other users to be able to locate him when he goes to see his doctor. This is simply a privacy problem. However, it is also possible to implement a rule in CBAC using two atrributes: a requester and Pipe's location.

If the same rule has been implemeneted in both CBAC and a privacy profile, the effect on the final decision of a resource manager might not be different, because the resource manager will make a decision such that the utility is the highest. Since there is a QoS component in the privacy profile, the resource manager can make use of that information. However, if a rule that should be implemented in a privacy profile is only implemeneted in CBAC, the resource manager will not be able to use the QoS component, not available in CBAC, to decide the best course of action.

Chapter 8

Contributions

8.1 Understanding access control issues in intelligent environment

Our biggest contribution in this thesis is the understanding of issues that might come up when a user tries to request a resource from others in various environment settings. While many research papers cover the details of access control models for intelligent environments, most of them focus on scenarios where there is only one user or multiple users in one space. However, interactions in intelligent environments can involve multiple users from multiple spaces, where each user can belong to different organizations. As a result, these models will not function adequately in the real world.

By analyzing intelligent environments in various settings, we have identified a few important key ideas that access control in intelligent environments should have: contextual information and privacy profile. Contextual information is necessary, because it allows access control rules to be expressive and be able to cover complex cases that can happen in intelligent environments. Another important point is that privacy is subjective and dynamic. It is subjective because each person has different tolerance. It is dynamic because users might change their preferences based on their experiences. As trivial as these facts seem, other access control mechanisms try to create rules that address both security and privacy at the same time. This approach makes rules more

complex, ridig, and less compatible with users.

8.2 New approach in designing access control for intelligent environments

The other contribution in this thesis is a new approach in designing access control for intelligent environments. Two important decisions that we have made are separating security and privacy and providing the framework to support QoS.

As discussed in the previous section, privacy is subjective and dynamic. As a result, privacy issues should not be bundled with security in an access control rule. We separate those two issues apart by designing rules that focus on security component and creating a privacy profile for each user.

In real world situations, privacy or security can be traded for QoS. As a result, we provide a framework to address this issue by making a privacy profile output QoS that will be taken into account by a resource manager when computing the overall utility.

Bibliography

[1] William Adjie-Winoto, Elliot Schwartz, Hari Balakrishnan, and Jeremy Lilley. The design and implementation of an intentional naming system. In *Proc. 17th SOSP*, pages 186–201, December 1999.

[2] John Barkley. Application enginnering in health care. In *Second Annual CHIN Summit*, 1995.

[3] M. Burnside, D. Clarke, T. Mills, A. Maywah, S. Devadas, and R. Rivest. Proxy-Based Security Protocols in Networked Mobile Devices. In *Proceedings of SAC 2002*, 2002.

[4] Ramaswamy Chandramouli. Application of xml tools for enterprise-wide rbac implementation tasks. Technical report, National Institute of Standards and Technology, 2000.

[5] Michael Coen. Design Principles for Intelligent Environments. In *Proceedings of AAAI'98*, 1998.

[6] Ferraiolo, Bugini, and Kuhn. Role based access control: Features and motivations. In *Computer Security Applications Conference*, 1995.

[7] Krzysztof Gajos. Rascal - a Resource Manager For Multi Agent Systems In Smart Spaces. In *Proceedings of CEEMAS 2001*, 2001.

[8] Krzysztof Gajos. Delegation, Arbitration and High-Level Service Discovery As Key Elements of A Software Infrastructure For Pervasive Computing, 2002. In submission.

[9] Nicholas Hanssens, Ajay Kulkarni, Rattapoom Tuchinda, and Tyler Horton. Building agent-based intelligent workspaces. In *ABA Conference Proceedings*, July 2002.

[10] Ajay Kulkarni. A Reactive Behavioral System For The Intelligent Room. Master's thesis, MIT, 2002.

[11] Marc Langheinrich. Privacy by Design - Principles of Privacy-Aware Ubiquitous Systems. In *In Proceedings of Ubicomp*, pages 273–291, 2001.

[12] Adam MacBeth. An Autoconfiguring Server-based Service Discovery System. Technical report, University of Washington, May 2001.

[13] Michael Covington Matthew. Generalized Role-Based Access Control for Securing Future Applications.

[14] Al Muhtadi, Anand Ranganathan, Roy Campbell, and N. Dennis Mickunas. A Flexible, Privacy-Preserving Authentication Framework for Ubiquitous Ccomputing Environments. In *Proceedings of IWSAEC 2002*, 2002.

[15] After Desktop Computing: A Progress Report on Smart Environments Research. *IEEE Intelligent Systems*, 15, 2000.

[16] Sofia K. Tzelepi, Dimitrios K. Koukopoulos, and George Pangalos. A Flexible Content and Context-based Access Control Model for Multimedia Medical Image Database Systems. In *Proceeding of ACM MMSig'01*, Ottawa, Canada, 2001.