Generalized Role-Based Access Control for Securing Future Applications[*]

Michael J. Covington[†] *Matthew J. Moyer* *Mustaque Ahamad*

College of Computing
Georgia Institute of Technology
Atlanta, Georgia

Abstract

As computing technology becomes more pervasive and broadband services are deployed into residential communities, new applications will emerge for the home and community environment. These applications will assist people in a variety of daily activities by enabling them to create, access, and manipulate information about the residents and resources in their homes. In a connected community, resources in the home and information about the residents of the home will be remotely accessible to both residents and guests, as well as to potentially malicious users. These new applications, as well as their users and environment, pose new security challenges. The challenges stem from two factors: the nature of the home itself—a private space with a wealth of personal and sensitive information—and the limited technical knowledge and capabilities of the home's residents.

We are addressing the problem of securing applications that will access and control information resources in the home of the future. Specifically, we are designing a security system based on a paradigm called *Generalized Role-Based Access Control (GRBAC)*. GRBAC is an extension of traditional Role-Based Access Control (RBAC). It enhances traditional RBAC by incorporating the notion of *object roles* and *environment roles*, with the traditional notion of *subject roles*. These new types of roles allow one to define rich, easy-to-understand security policies without having significant technical knowledge of the underlying computer systems that implement those policies. In this paper, we motivate the need for GRBAC, provide a high-level description of it and demonstrate its usefulness and flexibility via several example applications.

1 Introduction

As computers become more common in the home and broadband technology is introduced into residential communities, new applications will allow a wide range of human activities (e.g., education, entertainment, social and community gatherings, etc.) to be conducted over the Internet. Such applications often will use information about the residents of homes, as well as various resources inside the home. Furthermore, these applications will access this sensitive information from many different locations. Therefore, the protection of private information about each home's resources and residents is a critical concern that must be addressed before such applications can be successfully deployed. For an "always connected" home, in which the residents may have very limited computer skills, this is clearly a challenging task.

From a security perspective, the home is a novel environment for exploring policies that are intended to control access to sensitive resources and data. We expect many future homes to feature a rich computation and communication infrastructure that includes a variety of sensors, such as video cameras and audio microphones. The information gathered by these sensors may be used to facilitate rich social interactions via the Internet. The sensors will record, manipulate and store information about the home's residents and their activities. Clearly, such information is private and must be protected from unauthorized access. Traditionally, homes have been secured by physical devices (e.g., burglar alarms, deadbolts, motion-activated lighting systems, etc.) that make

[*] This work was completed at the Georgia Institute of Technology's College of Computing in Atlanta, Georgia. The research of Michael J. Covington was supported by an Intel Foundation Graduate Ph.D. Fellowship. The work of Matthew J. Moyer and Mustaque Ahamad was supported in part by the industrial partners of the Georgia Tech Broadband Institute and NSF grant CCR-9619371.

[†] Contact Author – `Michael.Covington@cc.gatech.edu`

it difficult for intruders to gain access into the home. But in the near future, when homes have networked information appliances that are accessible via the Internet and so-called "intruders" can enter the home digitally, these physical mechanisms will offer little or no protection from these "virtual" attacks.

Leaving homes vulnerable to electronic "trespassers" is a legitimate concern that must be addressed before security-sensitive applications can be deployed. Clearly, any compromise of the system entails the possibilities of data theft and mass distribution of private information. Unlike a physical burglar, an electronic intruder can attack the home at any time, from any location. Data such as tax or medical records, the whereabouts of family members, and even the hours of the day during which the home is unoccupied, are sensitive and private, and should be protected with at least the same level of security as any physical security devices can provide.

At Georgia Tech, we are building a prototype "home of the future" that will be used to explore a variety of new applications. These applications range from remote control of appliances in the home (e.g., Cyberfridge [9]) to awareness and monitoring of each resident's activities and needs. For example, researchers are exploring how a "smart" home can help elderly residents by monitoring their medical condition. The prototype home will have a rich computation and communication infrastructure and will be connected to other homes and institutions in the community. A variety of sensors will be used to infer the activities of the home's residents, and various applications will use this information to help improve their quality of life. This experimental home provides an excellent context in which to explore the security needs of future applications and the system support necessary to secure them.

Although considerable work has been done in securing military and commercial information systems, few projects have specifically addressed the needs of a residential computing infrastructure. We are developing security techniques that can be used in the home and community environments. More precisely, we are exploring access control models and security policies that can be used to secure next-generation home applications easily and intuitively. Our access control model is called *Generalized Role-Based Access Control (GRBAC)*. GRBAC builds upon traditional Role-Based Access Control (RBAC) [4, 13] with two new concepts: *object roles* and *environment roles*. This extension unifies ideas from several existing access control models into one elegant model that captures all security-relevant state in a system. The unification of all security-relevant state into a single concept—that of a role—makes access control policies significantly easier to define and implement in GRBAC than in other models. Ease of security policy definition and implementation is a key requirement for emerging applications in the home and community, since residents in the typical home are not experts in either computers or security.

The remainder of this paper proceeds as follows: in Section 2, we briefly describe Georgia Tech's Aware Home project and the applications being explored there within. Then, in Section 3, we discuss the security challenges presented by this environment. To meet these challenges, we have developed the GRBAC model, which we describe in Section 4. In Section 5, we present examples of several applications and show how GRBAC can be used to specify their security needs. We discuss related work in Section 6, and we conclude the paper in Section 7.

2 The Aware Home

The Aware Home project at Georgia Tech is the creation of an interdisciplinary team of researchers who wanted to build an information-rich "home of the future" from the ground up [7]. When completed, this home will have a rich computation and communication infrastructure, including a variety of sensors and cameras that will make the home "aware" of its residents and their activities [10]. Various applications will exploit this "awareness" to make daily living easier for the home's residents. The new applications that will be explored in the Aware Home relate to the domains of education, entertainment, physical security, inventory management (e.g., for the contents of the pantry or refrigerator), and utility management (e.g., for gas and electricity), as well as applications that permit rich interactions with people and institutions outside the home.

Consider some specific examples of Aware Home applications. One research group is exploring how the Aware Home concept can help elderly residents remain in their homes longer, rather than having to move into assisted living communities. This application uses the home's sensors to enable important interactions with relatives outside of the home and with care specialists, effectively providing the same level of care and supervision that today can be found only in nursing homes and hospitals. Another class of applications will allow residents to manage inventories in the home from any location, inside or outside the home. For example, the Cyberfridge application [9] collects information about food items in a refrigerator and makes the data accessible from anywhere. Cyberfridge can interface with a local food delivery service to automatically reorder food items such as milk or eggs when necessary. A third example is an application that automatically manages home resources such as hot water and heat, based on the residents' preferences and daily living habits. It can, for example, choose to heat the house only when it knows there are residents inside, and it can choose to produce hot water only at times when residents usually take showers or wash clothing. Such an application can even negotiate the best possible electricity rates with utility services, based on the needs and preferences of the home's residents.

All of the applications described above share a common thread: they are possible as a result of the Aware Home's ability to gather, store and transmit useful information about the state of its resources and occupants. Clearly, this information should be available only to legitimate users and applications. Financial loss, public embarrassment and even physical harm are just a few of the many potential negative consequences of a breach in the security of any of these applications. Therefore, from a security standpoint, the Aware Home project presents a unique opportunity to explore the support that is necessary to secure the home of the future, before the technology—and the risk that accompanies it—become widely available. The next section discusses security issues in the home in greater detail.

3 Security Challenges in the Home

The Aware Home applications described in the previous section present new and interesting security challenges. First, the residents of a home usually know little about information security or computer technology. Despite their lack of expertise, however, they often will need to configure and manage information security policies in their homes. Therefore, system usability is critical. In short, the system must make it *very easy* for a homeowner to define and manage security policies for the applications and resources in the home; otherwise, it is likely the homeowner will not use the security features of the system at all. This requirement encompasses all aspects of usability, including learnability and the generation of appropriate feedback to assure the user that she is using the system correctly. Second, the system must not intrude upon the everyday activities of residents in the home. For example, it is unreasonable to expect a resident to explicitly authenticate herself to the home security system before opening the refrigerator. Essentially, the security mechanisms must be both *usable* and *non-intrusive*, or many homeowners will simply avoid using them. In the remainder of this section, we explore these requirements in more detail.

Security policies in the Aware Home potentially can be quite complex, as we will demonstrate via the following examples. A policy can constrain access to information or resources based on several factors, including attributes about the subject, the resource or the environment. For example, subjects can be classified as "resident" or "guest," "adult" or "child," or even as a "pet." Access rights then can depend on the subject's attributes (e.g., "resident"), as well as on his or her identity. Access also may be constrained based on the subject's location, or based on environmental factors such as the temperature or the time of day. For example, a policy might say that a repairman has access to the refrigerator only while he is inside the home on January 17, 2000, between 8:00 a.m. and 1:00 p.m. In addition, access to information objects or resources may depend on security-relevant attributes of the object's state. For example, a child may be prohibited from viewing any television program or movie that is not rated "G" or "PG". As a final example, both positive and negative access rights arise naturally in the context of the home. For example, adult residents may be granted access to all appliances in the home, while children are denied access to potentially dangerous appliances. Given these simple examples, it is easy to understand how any access control system for the home must be both flexible and easy to use. In subsequent sections of this paper, we will describe why we believe that the GRBAC access model fulfills these requirements.

Another challenge presented by the home environment is relieving the user from the burden of access control decisions. Ideally, information available from sensors in the home should be used to automatically infer a subject's security-relevant attributes (e.g., identity, role or location.) For example, several technologies such as voice and face recognition are being deployed in the Aware Home to non-intrusively identify humans and track their movements. Many such techniques can establish the identity of a subject with only a partial level of certainty, or *confidence level*. Such "partial authentication" has important implications for access control models. In particular, some identification mechanisms are known to be more reliable than others. For example, an experiment might conclude that face recognition is 90% accurate, while voice recognition is only 70% accurate. This introduces a potential problem regarding the identification of subjects in the home. If one type of sensor can identify a subject with a higher degree of accuracy than another, then the system should permit the definition of security policies that account for the difference in accuracy. For example, consider an adult who wants to view the output of a video camera in a child's bedroom, for the purpose of checking on the child. The security policy may state that only the child's parents or babysitter can view the video. Perhaps a "strong" identification mechanism may provide enough authentication evidence to allow the user to see a streaming video, while a "weak" identification mechanism may provide only enough authentication evidence to permit the user to view a recent still image of reduced quality and definition. A security model for the home should incorporate these confidence levels for both authentication and access control. In the following section, we introduce GRBAC, an access model that we believe can fulfill these challenging requirements.

4 Generalized Role-Based Access Control

In the previous section, we discussed some of the challenges facing a security system for an Aware Home environment. We are designing a system that we believe will meet these challenges. At the core of this system is our Generalized Role Based Access Control (GRBAC) model. GRBAC is an extension of traditional Role Based Access Control (RBAC) that uniformly applies the concept of roles not only to subjects, but also to objects and system states. In this section we introduce GRBAC. First, we review the most important features of traditional RBAC. Then, we discuss the fundamental concepts of GRBAC and describe some of the important issues relating to it.

4.1 Traditional RBAC: A Foundation for GRBAC

Traditional Role Based Access Control (RBAC) [4, 13] is a form of mandatory (i.e., centrally administered) access control. It is based on the premise that most real-world access control decisions are determined by a person's position or job title within an organization. Accordingly, the purpose of RBAC is to encourage the design of security policies that closely mirror the structure of organizations. In this section, we highlight the most important features of the traditional RBAC model.

4.1.1 RBAC: Basic Features and Rules

The basis of RBAC is the concept of a *role*. Fundamentally, a role is a grouping mechanism that is used to categorize subjects based on various properties. Much of the RBAC model is based on the mathematics of set theory; thus, many of the constructs of the RBAC model are based on the notion of set membership. Individual users in an RBAC system are called *subjects*. Each subject has an *authorized role set*, which consists of all the roles that the subject has been authorized to use. We use the term *role possession* to denote that a role is in the authorized role set of a subject. In other words, we say that subject S *possesses* role R if S has been authorized to use R.

The other two fundamental concepts in RBAC are the *object* and the *transaction*. An object is any resource in a system. Example resources in the home include appliances such as a dishwasher or stereo, media objects such as movies, and sensitive digital information such as medical records or income tax returns. A transaction is a series of one or more accesses to a set of one or more objects. A transaction in the home could be as simple as reading file `foo` on the family computer. In contrast, a transaction in a military setting can be as complex as aiming and firing a missile from a Navy destroyer. All policy rules in RBAC are linked to roles, rather than to individual subjects. Formally, each role R is associated with an *authorized transactions set*; this is the set of transactions that a subject may perform using role R. Therefore, for a subject S to gain access to transaction T, S must demonstrate possession of a role R, for which T is in the authorized transactions set of R. Figure 1 summarizes the basic RBAC features.

4.1.2 RBAC: Some Problems and Solutions

At its core, RBAC is quite simple; however, in practice, RBAC policies can become very complex and unwieldy. In this section, we describe some of the problems that RBAC systems face, as well as several advanced RBAC features that have been used to solve the problems. We first examine two problems that stem from the complexity of policies: separation of duty and role precedence. Then we discuss role activation and role hierarchies, two constructs that can help mitigate these problems.

Separation of Duty It is implicit from the previous section that a subject can possess multiple roles simultaneously. Typically, there are no problems associated with multiple role possession; however, there are some

Definitions:

Subject $\mathcal{S}$	a user of the system
Role $\mathcal{R}$	a categorization primitive for subjects
Object $\mathcal{O}$	a system resource
Transaction $\mathcal{T}$	a series of one or more accesses to one or more objects
$AR(\mathcal{S})$	the authorized role set for subject $\mathcal{S}$
$AT(\mathcal{R})$	the authorized transaction set for role $\mathcal{R}$
$\texttt{exec}(\mathcal{S}, \mathcal{T})$	*true* iff subject $\mathcal{S}$ is authorized to execute transaction $\mathcal{T}$

RBAC Access Mediation Rule:

$\texttt{exec}(\mathcal{S}, \mathcal{T})$	*true* iff $\exists$ role $\mathcal{R}$: $\mathcal{R} \in AR(\mathcal{S})$, $\mathcal{T} \in AT(\mathcal{R})$

Figure 1: Basic RBAC Definitions and Rules

cases in which the set of access privileges granted by multiple role possession can constitute a conflict of interest. For example, in a financial institution, two possible roles are teller and account holder. An account holder authorizes certain actions (such as withdrawals and deposits) on his account, and a teller executes those actions. If a person is a bank employee and also owns a checking account at the bank, there exists the potential for that person to act as both an account holder and a teller at the same time. With the privileges of both account holder and teller, an employee may be able to perform illegal actions, such as making fraudulent deposits, on his account.

Such scenarios occur often in RBAC systems; the circumstance described above is known as a *separation of duty* problem. There are two varieties of separation of duty: *static* and *dynamic*. Dynamic separation of duty occurs when two roles present a conflict of interest if a subject uses them both at the same time. The conflict of interest described above is an instance of dynamic separation of duty. Note that there is no conflict of interest if the employee acts as a teller during one time interval and an account holder during another interval, since only when he assumes both roles simultaneously is it possible for him to abuse the system. In contrast, static separation of duty occurs when two roles present a conflict of interest that cannot be resolved by simply preventing the roles from being used simultaneously. In these cases, the two roles may *never* be used by the same subject. If roles *R1* and *R2* exhibit static separation of duty, and subject *S* has acted in role *R1*, he may never act in role *R2*.

Role Precedence Another problem that relates to multiple role possession is *role precedence*. Role precedence stems from inconsistent access rules between two roles that a subject possesses. For example, in the home environment, suppose that user *Bobby* is authorized to possess the roles of *child* and *family member*. Suppose also that the *family member* role is authorized to read family medical records, but the *child* role is not. If *Bobby* tries to read the family's medical records, the system must decide how to resolve the inconsistency in the access policy. To solve the problem, the system must decide which access rule takes precedence over the other. There are many ways to make this decision. The simplest way would be to always give precedence to the role that denies access. Similarly, the system could always give precedence to the role that allows access. Or there could be some other predefined rule or algorithm established to decide role precedence. One approach to solving this problem, as we discuss below, is the use of *role activation*. Role precedence is a problem that every RBAC system must solve.

Role Activation We discussed above the concept of an authorized role set: the set of roles that a subject is allowed to use. The problems of separation of duty and role precedence both are related to an authorized role set, because as the size of an authorized role set grows, separation of duty and role precedence become more difficult to manage. One of the most common and effective ways to handle this problem is to restrict a subject's role usage to a subset of his authorized role set at all times, so that only those roles that are necessary to perform his current duties are *active*. This is called *role activation*. When role activation is used, a subject must declare which roles he intends to use at all times. The roles that have been declared active constitute the subject's *active role set*. Only roles in the active role set can be used to execute transactions. This mechanism allows the system to more easily enforce dynamic separation of duty constraints: the system simply disallows any two roles with dynamic separation of duty constraints from being active at the same time. Role activation also provides a natural mechanism for resolving role precedence: in case of a conflict between roles, active roles take precedence over inactive roles.

Role Hierarchies Another useful RBAC construct is the *role hierarchy*. Role hierarchies can help manage role complexity through structure to exploit commonality not only among subjects but among roles as well. For example, in an organization all managers may have a certain set of core "management privileges" even though they all work in different departments. This commonality can be exploited through a role hierarchy that makes each department manager role a sub-role of a generic "managers" role. Role hierarchies allow a policy implementor to write generic access rules just once, rather than once for every role to which the rules apply. This kind of structuring tool can help avoid policy "bugs": cases in which the policy implementor has incorrectly written the policy. Hierarchies also can serve as a tool for cleaner policy design, thereby eliminating some cases in which role precedence conflicts might otherwise have occurred.

4.2 The GRBAC Model

Traditional RBAC is very useful, but it suffers from subject-centric limitations that restrict the policy designer to a subject-oriented viewpoint. Generalized Role Based Access Control (GRBAC) is an extension of RBAC that removes the subject-centric limitation, allowing a policy designer to write the policy from a subject-centric, object-centric, or environment-centric viewpoint, or whatever combination of these is most appropriate for the circumstances. GRBAC removes the limitations of RBAC by using the basic concept of a role to organize *all* entities in a system. It exploits the organizational power of roles for grouping environment states and objects, in addition to subjects. This section introduces our GRBAC model at an informal level. Interested readers are encouraged to consult [11] for a more formal treatment of the GRBAC model.

4.2.1 Subject Roles

A *subject role* in GRBAC is analogous to a traditional RBAC role. Each subject is authorized to assume a set of subject roles. Subject roles may be hierarchical or "flat" (single-level) in nature. The system may also use subject role activation. The only difference between GRBAC subject roles and traditional RBAC roles is the way that they are used to make access decisions. In traditional RBAC, an access decision is based entirely on the permissions associated with the set of roles that the subject possesses. In GRBAC, an access decision depends not only on subject roles, but also on environment roles and object roles. We describe each of these roles below.

4.2.2 Environment Roles

There are many real-world instances in which access control depends not only on the person making the access and the object being accessed, but also on the state of the environment during the access. For example, many organizations restrict access to their facilities during nights and weekends. In the military, secure computer systems are often restricted only to personnel in designated physical areas, such as a highly secure computer room. In the home, parents might restrict their children's access to the television, allowing the kids to watch TV only after they have done their homework, and only until 9:00 p.m. In each of these instances, the access control policy depends on information from the *environment*. The two most basic types of environmental information are time and location, but any security-relevant information in the environment that can be accurately captured by the system also can be used to control access to system resources.

The GRBAC model allows policy designers to specify system state through *environment roles*. An environment role can be based on any system state that the system can accurately collect. For example, we can define a role corresponding to each day of the week, or each month of the year. A policy rule such as "managers may edit salary data for their employees only on the first Monday of each month" is easy to implement using environment roles. Similarly, environment roles may be used to describe rules that relate access permissions to the locations of subjects. In the home, we can define location roles such as "upstairs," "downstairs," "master bedroom," etc. We can then use these roles in policy rules; for example: "children may only use the videophone while they are in the kitchen."

Relating to environment roles, there clearly are many tangential issues that must be addressed before environment roles can be used in real access control systems. First and foremost, the system must be able to securely and accurately collect enough system data (e.g., an accurate estimate of the current time, or the location of a subject in the home) to determine whether a given environment role is active. One effective approach to this problem would be to use a trusted event system that is capable of generating events based on various system state changes. Second, the system must provide a means for policy implementors to define roles. Some basic environment interface must exist, so that policy writers can associate their environment role definitions with actual system states. Both of these issues are the subject of ongoing research and are beyond the scope of this paper.

4.2.3 Object Roles

Subject roles and environment roles allow a policy implementor to structure a policy based on either the properties of the subjects in the system, or the system state itself. But what if the policy implementor wants to structure the policy according to the properties of the *resources* in the system? To accommodate this scenario, the GRBAC model also includes *object roles*. Object roles allow us to capture various commonalities among the objects in a system, and use these commonalities to classify the objects into roles. Object roles can be based on any classifiable property of an object, including its date of creation, object type (image, source code, streaming video, etc.), sensitivity level (secret, top secret, etc.), or information about the contents of the object (for example, we could classify objects based on whether they contain any content related to Microsoft Corporation). After classifying the objects, we can make access control decisions based on the classification scheme that we created.

4.2.4 Making Access Decisions with GRBAC

In Figure 1, we outline the basic algorithm for mediating access to objects in the traditional RBAC model. In RBAC, if subject S wants to access object O, S must possess a role R that is authorized to execute transaction T, such that T can access O. In GRBAC, the access mediation algorithm is similar, but slightly more complex. Subject S possesses a set of subject roles, and object O possesses a set of object roles. In addition, the system keeps track of a set of environment roles. For S to access O, S must possess some subject role R_S, such that:

1. $\exists$ some object role R_O, possessed by O;
2. $\exists$ some environment role R_E that is currently active;
3. $\exists$ some transaction T that allows R_S to access objects in role R_O when R_E is active.

Clearly, this access mediation rule is more complex than the corresponding rule for traditional RBAC.

In Section 4.1.2, we briefly discussed separation of duty and role precedence, two of the potential problems that can arise in an RBAC system. These two problems are not confined to traditional RBAC; they also can cause difficulty in the GRBAC model. In fact, GRBAC's generality makes it even more susceptible to various types of policy conflicts and ambiguities. Our purpose in this paper is not to outline all of these potential problems, but simply to introduce the reader to the fundamental GRBAC concepts of subject roles, object roles and environment roles. We do not discuss the GRBAC model in any more detail here; however, we encourage interested readers to refer to [11] for a more thorough review of the model, its usage, and the problems that can arise from it.

5 GRBAC in Practice

GRBAC is a powerful and elegant model for specifying access control rules in a computationally rich environment. This section shows how GRBAC can be applied in practice to the home environment. It also illustrates some of the additional security benefits that GRBAC can provide in a system.

5.1 A Simple Example

To illustrate the power and elegance of GRBAC, we begin by creating a subject role hierarchy, such as the one displayed in figure 2. This role hierarchy presents a graphical view of the sample household that we will consider in the following scenario. Specifically, it shows the relationships that exist between the various users and the roles that are present in the system. The figure shows that users *Mom* and *Dad* have each been assigned to the *Parent* role. In addition, users *Alice* and *Bobby* have been assigned to the *Child* role. The system also can accommodate an authorized household guest, *Dishwasher Repair Technician*.

Assume that *Mom* and *Dad* have decided to permit the children to use entertainment devices (such as the tele-

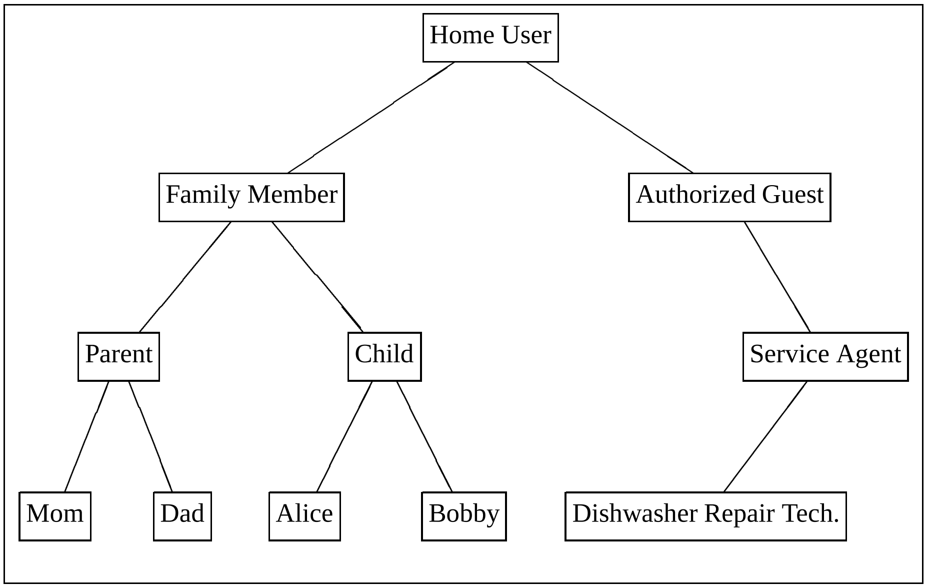

Figure 2: An Example Subject Role Hierarchy for the Home

vision and VCR) on weekdays, but only during their free time after dinner, before going to bed. To enforce this policy, the system must be configured to identify the various entities in the system and classify them into the particular roles that are relevant to the access decision being processed.

In this particular example, the system must use all three types of GRBAC roles. First, the users must be classified so that a specific user identity can be mapped to a role, such as *Parent* or *Child*. By mapping users to roles, the home administrator can specify an access control policy for a *group* of users, rather than for each individual user. If the access policy requires the administrator to classify a subset of users from both the *Parent* and *Child* roles, he can simply create a new role and map users into it as necessary.

In addition to subject roles, the system in this example uses an environment role named *weekdays*. *Weekdays* are defined by the system as the time from 12:01 a.m. on Monday to 11:59 p.m. on Friday. Also, since dinner usually is over by 7:00 p.m., and since the children have a bed time of 10:00 p.m., the environment role *free time* is defined to be 7:00 p.m. to 10:00 p.m.

Finally, the system has a defined object role named *entertainment devices*. Objects that map to this role include all televisions, stereos and home video games. If the household were to purchase a new toy or entertainment device, they could simply map the device to the role and it would immediately be controlled by this predefined access policy.

After defining all the necessary roles, the administrator needs to write just one rule to implement the policy. The rule in this case is “any *child* can use *entertainment devices* on *weekdays* during *free time*.” This example shows how GRBAC makes it easy to take a fairly complex access policy and implement it cleanly and efficiently using subject, object, and environment roles.

5.2 Enhancing a System with GRBAC

The scenario above presents some sample GRBAC roles and illustrates how role relationships can be used to establish security policies for the home. As stressed earlier, ease of security policy definition and implementation is a key requirement for applications in this domain, because we cannot require all homeowners to fully understand information security. In addition, we have stated that the system and related security mechanisms must be non-intrusive and easy to use. In this section, we briefly explore how GRBAC can be used to enhance a system and also fulfill these requirements.

As discussed in Section 3, a system should make access decisions without placing any undue burden on the users. Unfortunately, access control without authentication is usually impossible. In the home, it is generally unacceptable to require users to explicitly “log in” before using a device or service. Instead, they should be identified implicitly by sensors throughout the home. These identification technologies are not perfect, however, and often may provide only "partial authentication" of users based on limited sensory information. Fortunately, GRBAC makes it possible in many cases to make access decisions based on only partial authentication information.

Consider the following scenario. Suppose that *Alice* is 11 years old and weighs 94 pounds. She has finished eating her dinner, and she wants to watch television before going to bed. As she approaches the television, the Smart Floor [12] can identify her as *Alice* with 75% accuracy by comparing the amount of weight that it senses with its internal, "official" weight for *Alice*, 94 pounds. Suppose that the security policy requires a person to be identified with 90% accuracy before the system will grant rights to that person. Based on this policy, *Alice* should not be allowed to access any resources, because the system has received insufficient data to authenticate her at the required 90% level. But given the structure of a GRBAC security policy, the system is not limited to making access decisions based only on a specific user's *identity*. The policy specifies that anyone in the *Child* role can use *entertainment devices* (an object role possessed by the television) during *free time*. Despite the fact that the Smart Floor is able to identify *Alice* with only 75% accuracy, it may be able to authenticate her into the *Child* role with 98% accuracy, because it knows (for example) the approximate weight of children in the household. Since the system can authenticate *Alice* into the *Child* role with higher than 90% accuracy, and there is a policy rule stating that children can use entertainment devices during free time on weekdays, it will grant her access to the TV when she pushes the TV power button.

6 Related Work

In this section, we briefly highlight several existing access models and compare them to GRBAC. We discuss traditional RBAC, time-based authorization, system-load-based authorization, content-based access control, and several other notable models. GRBAC allows us to express policies supported by these other models, and it also provides an elegant means of unifying all of their major concepts.

We discussed traditional RBAC [4, 13] in Section 4. Traditional RBAC is essentially GRBAC with subject roles only. The GRBAC model is more versatile and more expressive than traditional RBAC because it uses not only subject roles, but environment roles and object roles as well. GRBAC clearly is a more complex model than RBAC, but we believe that with appropriate care for "clean" (i.e., well-structured) policy definition and the right set of constructs for creating "clean" policies, the additional expressive power provided by GRBAC justifies its additional complexity.

Bertino et al. [2, 3] have investigated support for temporal authorizations in database systems. They have examined both periodic and non-periodic authorizations. Their access control model is discretionary, whereas GRBAC is mandatory. But in principle, their notion of temporal authorization is similar to GRBAC's notion of time-based environment roles. We believe our model is better than theirs in terms of its usability and flexibility. In GRBAC, environment roles can be used to simplify temporal access rules by assigning human-understandable names to various periods of time, e.g., "Monday," "Weekends," or even "Weekday mornings in July." In contrast, their authorization language is very technical, which inherently limits its usefulness to the small set of people who have the background necessary to understand it.

Similarly, in their Generalized Access Control Language (GACL), Woo and Lam [15] use the notion of *system load* as a determining factor in certain access control scenarios, so that, for example, certain programs only can be executed when there is enough system capacity available to handle them adequately. Given appropriate support for monitoring and reporting changes in system state, the GRBAC model can also support such state-based authorization decisions using environment roles. In fact, the scope of GRBAC environment roles is limited only by the level of support that the system provides for accurately reporting environmental information.

Gopal and Manber [6] discuss how to integrate content-based access mechanisms into traditional file systems. Their work is based on Gifford's Semantic File System [5]. GRBAC also supports a form of content-based access control using object roles; however GRBAC differs slightly from their model in its focus. Specifically, they focus on the systems issues involved in efficiently integrating hierarchical file systems with database-like query functionality. In contrast, GRBAC focuses on elegance, clarity of concept, and usability.

There are several other access control models that are worth noting due to their influence on the design of GRBAC; we briefly mention them here. The first related model is multilevel access control [1], which traditionally has been used in military computer systems for highly sensitive data. Its basic premise is to allow information to flow up the chain of security levels, but never down. The GRBAC model can be used to implement multilevel access control, but the converse is not true. Another related access control model was proposed by Jajodia et al. [8]. It seeks to separate access policy from access mechanism by providing the policy designer with a language that is provably capable of expressing any access policy. Finally, we note the work of Shen and Dewan [14]. They have developed a flexible, powerful role-based model for access control in collaborative environments, where policies must account for concurrent operations on shared objects and other complex access issues.

7 Conclusion

In this paper we have introduced a new access control model, Generalized Role-Based Access Control (GRBAC), and described why we believe it will be useful for securing applications in the highly-connected homes of tomorrow, as well as in other environments. The major benefit of GRBAC over current access control models is its combination of usability and expressiveness. GRBAC is easy to use because it is based on one main concept: the *role*; however, the uniformity and flexibility with which roles are applied to subjects, objects and environment states also makes the model very powerful and expressive. It is important to note that GRBAC is not a complete security solution in itself. It is only an access control *model*; to be useful in the real world, it must be integrated carefully into a trusted computer system. In the near future, we intend to explore these integration issues and build a prototype system based on GRBAC.

References

[1] D. Elliott Bell and Leonard J. LaPadula. Secure computer systems: Mathematical foundations. Technical Report MTR-2547, The MITRE Corporation, November 1973.

[2] Elisa Bertino, Claudio Bettini, Elena Ferrari, and Pierangela Samarati. Supporting periodic authorizations and temporal reasoning in database access control. In *22nd VLDB Conference*, 1996.

[3] Elisa Bertino, Claudio Bettini, Elena Ferrari, and Pierangela Samarati. A temporal access control mechanism for database systems. In *IEEE Transactions on Knowledge and Data Engineering*, volume 8, 1996.

[4] David F. Ferraiolo, John F. Barkley, and D. Richard Kuhn. A role based access control model and reference implementation within a corporate intranet. In *ACM Transactions on Information Systems Security*, volume 1, February 1999.

[5] David K. Gifford, Pierre Jouvelot, Mark A. Sheldon, and James W. O'Toole. Semantic file systems. In *Proceedings of ACM SIGOPS Symposium on Operating Systems Principles (SOSP)*, October 1991.

[6] Burra Gopal and Udi Manber. Integrating content-based access mechanisms with hierarchical file systems. In *Operating Systems Design and Implementation (OSDI) Sympsoium*, 1999.

[7] The Future Computing Environments Research Group. The Aware Home: The BTC/FCE experimental house. Research Group Web Page, 1999-2000. http://www.cc.gatech.edu/fce/house/.

[8] Sushil Jajodia, Pierangela Samarati, V. S. Subrahmanian, and Elisa Bertino. A unified framework for enforcing multiple access control policies. In *Proc. of the 1997 ACM International SIGMOD Conference on Management of Data*, May 1997.

[9] Jennifer Mankoff and Gregory Abowd. Domisilica: Providing ubiquitous access to the home. Technical Report GIT-GVU-97-17, College of Computing, Georgia Institute of Technology, May 1997.

[10] D. Moore, I. Essa, and M. Hayes. Exploiting human actions and object context for recognition tasks. In *IEEE International Conference on Computer Vision*, 1999.

[11] Matthew J. Moyer and Mustaque Ahamad. Generalized role based access control. Technical Report GIT-CC-00-16, College of Computing, Georgia Institute of Technology, July 2000. Publication Pending.

[12] Robert Orr, Gregory Abowd, Chris Atkeson, Irfan Essa, and Robert Gregor. The smart carpet: A mechanism for user identification and location tracking. Georgia Tech Graphics, Visualization, and Usability (GVU) Center Seed Grant, June 1998.

[13] Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman. Role based access control models. In *IEEE Computer*, volume 2, February 1996.

[14] Honghai Shen and Prasun Dewan. Access control for collaborative environments. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, pages 51–58, November 1992.

[15] Thomas Y. C. Woo and Simon S. Lam. Designing a distributed authorization service. In *Proceedings of IEEE INFOCOM*, March 1998.