

On Spatio-Temporal Constraints and Inheritance in Role-Based Access Control

Liang Chen
l.chen-2@rhul.ac.uk

Jason Crampton
jason.crampton@rhul.ac.uk

Information Security Group and Department of Mathematics
Royal Holloway, University of London

ABSTRACT

Pervasive computing environments have created a requirement for spatial- and temporal-aware access control systems. Although temporal, spatial and spatio-temporal role-based access control (RBAC) models have been developed, a family of simple, expressive and flexible models that convincingly addresses the interaction between spatio-temporal constraints and inheritance in RBAC does not yet exist. In this paper, we define three spatio-temporal models based on RBAC96 the *de facto* standard for RBAC, and extend these models to include activation and usage hierarchies. These models provide different authorization semantics, varying in the extent to which RBAC entities and relations are constrained by spatio-temporal restrictions. We introduce the notion of trusted entities, which are used to selectively override certain spatio-temporal restrictions. We also demonstrate that our spatio-temporal models are consistent and compatible with RBAC96 and the ANSI-RBAC standard, in contrast to existing models. Finally, we propose four approaches to encoding spatio-temporal requirements in practical applications that permit access requests to be answered efficiently.

Categories and Subject Descriptors

D.4.6 [Security and Protection]: Access controls; K.6.5 [Management of Computing and Information Systems]: Security and Protection

General Terms

Security, Theory

Keywords

ERBAC, RBAC, Spatio-temporal domain

1. INTRODUCTION

Role-based access control (RBAC) has been the subject of considerable research in the last decade [1, 10, 14] and is

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ASIACCS '08, March 18-20, Tokyo, Japan

Copyright 2008 ACM 978-1-59593-979-1/08/0003 ...\$5.00.

widely accepted as an alternative to traditional discretionary and mandatory access controls. The emergence of mobile and ubiquitous computing environments poses new demands on access control mechanisms, because the decision to grant access may depend on contextual information, such as the location of the user and the time at which access requests are made. It may be appropriate, for example, to limit the time and places at which a particular role can be activated.

Several context-based RBAC models have been defined in recent years [3, 4, 6, 7, 8, 9, 12, 15]. Each of these models introduces extensions to the basic role-based model in which components may be associated with general contextual constraints [6, 7, 15], temporal constraints [3, 9], spatial constraints [4, 8], or spatio-temporal constraints [12]. However, none of these models accurately captures the interaction between spatio-temporal constraints and inheritance in RBAC model: indeed, all of them have one or more of the following limitations.

- No existing model has clear semantics for inheritance in the role hierarchy in the presence of spatio-temporal constraints. This means that there is no way of designing an algorithm for deciding access requests.
- Existing models are extremely complicated. GTR-BAC [9] and the spatio-temporal RBAC model of Ray and Toahchoodee [12] define a large number of predicates to specify temporal constraints and spatio-temporal constraints, respectively. The relationship between these predicates is often unclear, again making it difficult to see how access requests should be evaluated in such models.
- Conflicts and ambiguity may occur in existing models. Conflicts may arise among the constraints specification in spatio-temporal RBAC [12], for example.
- Existing models lack compatibility with RBAC96 and the closely related ANSI-RBAC standard. It is not at all clear how to translate the predicates used in GTR-BAC [9] and spatio-temporal RBAC [12], for example, to the entities and relations used in the ANSI-RBAC standard and RBAC96.

In summary, we would argue that existing models focus far too much on syntax, and far too little on semantics.

In this paper, we undertake a rigorous analysis of spatio-temporal requirements in RBAC models. We use a graph-based formalism to define the semantics of RBAC96 [14] and ERBAC07 [5]; this formalism provides the basis for the semantics of our subsequent models. We then define three

different spatio-temporal RBAC models, motivating the development of these models using simple scenarios. These models extend the basic RBAC96 model with very little additional syntax. We also introduce the idea of trusted entities: for such entities spatio-temporal constraints may be ignored, in order to deal with certain scenarios. These simple, expressive, flexible spatio-temporal RBAC models have clear, well-defined semantics and are designed to be compatible with RBAC96 and the ANSI-RBAC standard.

In an effort to address interoperability problems that exist when there is a single role hierarchy and separation duty constraints, ERBAC96 [13] and ERBAC07 [5] have been defined to separate the role *activation* hierarchy and the permission *usage* hierarchy. We extend our models to include spatio-temporal requirements for ERBAC07.

The existence of spatio-temporal constraints will generally result in a more complex access control decision function. Hence, we consider the implementation of our spatio-temporal RBAC models in practical applications. We find that it is possible to use one of our models to encode most spatio-temporal requirements, if we assume that there is no role hierarchy. On the other hand, if a role hierarchy is required, we show how to eliminate spatio-temporal constraints on roles by imposing restrictions on outer nodes and edges, such as users and user-role assignments. In addition, we demonstrate how to improve the efficiency of access request checking by pre-computing spatio-temporal constraints over the transitive closure of (part of) the RBAC graph. We believe that our models can be efficiently and easily implemented, in contrast to existing models.

All existing models tend to focus on the syntax of temporal and spatial constraints, rather than on the semantics of the model. While we believe that the syntax of such constraints will generally be application-dependent, we do consider the representation of spatio-temporal constraints, distinguishing between concrete and symbolic domains. Concrete domains comprise a set of points defined by some numerically-encoded reference system; symbolic domains comprise sets of labels, each corresponding to one or more points in a concrete domain. Our approach is considerably simpler and more general than existing work, such as GTRBAC [9].

The rest of the paper is organized as follows. In the next section, we recall the basic features of RBAC96 and ERBAC07. We also introduce a novel graph-based formalism to define the semantics of RBAC96 and ERBAC07. In Section 3, we formally define the RBAC_{ST}^- , RBAC_{ST}^+ and RBAC_{ST}^- models, and introduce the notion of trusted entities. We also demonstrate the use of RBAC96-style syntax to encode spatio-temporal RBAC models, and illustrate how to integrate our spatio-temporal functions into the ANSI-RBAC standard. In Section 4, we introduce the ERBAC_{ST}^- , ERBAC_{ST}^+ and ERBAC_{ST}^- models for ERBAC07. In Section 5, we consider the practical considerations in spatio-temporal RBAC models, focusing on the interaction between spatio-temporal constraints and the role hierarchy. In Section 6, we discuss possible representations of spatial and temporal domains, and give concrete examples of spatial RBAC_{ST}^- , temporal RBAC_{ST}^- and spatio-temporal ERBAC_{ST}^- . Section 7 compares our work with related work in the literature. Section 8 concludes the paper with some suggestions for future work.

2. PRELIMINARIES

2.1 RBAC96 syntax

The RBAC96 family of models is undoubtedly the most well known model for RBAC [14], and provides the basis for the recent ANSI RBAC standard [1]. RBAC_0 , the simplest RBAC96 model, introduces a set of users U , a set of sessions S , a set of roles R , a set of permissions P ,¹ a user-role assignment relation $UA \subseteq U \times R$ and a permission-role assignment relation $PA \subseteq P \times R$. A user u may activate a role r in a session s if there exists $r \in R$ such that $(u, r) \in UA$. A user u is authorized for permission p if there exists $r \in R$ such that u may activate r and $(p, r) \in PA$.

RBAC_1 introduces the concept of role hierarchy, which is modeled as a partial order on the set of roles (R, \leq) . The role hierarchy $(RH \subseteq R \times R)$ is used to reduce the administrative burden by reducing the number of explicit assignments in UA and PA relations. That is, a user u may activate a role r in session s if there exists $r' \in R$ such that $(u, r') \in UA$ and $r \leq r'$. A user u is authorized for permission p if there exists $r, r' \in R$ such that u may activate r' , $(p, r) \in PA$ and $r \leq r'$.

RBAC_2 extends RBAC_0 through the addition of access control constraints, such as separation of duty constraints, cardinality constraints etc. RBAC_3 incorporates the features of RBAC_1 and RBAC_2 , although it has long been known that the features of those two models are somewhat incompatible. From now on, we write RBAC96 to mean RBAC_1 only; note that RBAC_0 is a special case of RBAC_1 in which the hierarchy relation is empty.

2.2 RBAC96 semantics

We introduce a novel graph-based formulation of RBAC96, which we believe to be simple and intuitive specification of basic components of RBAC96 model. As we will see, this formulation can be readily extended to include spatio-temporal restrictions. We construct an acyclic, directed graph $G = (V, E)$, where $V = U \cup R \cup P$, and $E = UA \cup PA \cup RH$. In other words, each vertex v represents an entity, such as a user u , a role r or a permission p in a RBAC96 system, and each directed edge $e = (v_i, v_j)$ represents a relationship between two entities v_i and v_j ; specifically, $(v_i, v_j) \in E$ if and only if (precisely) one of the following conditions holds

$$\begin{aligned} (v_i, v_j) &\in UA, \\ (v_j, v_i) &\in RH, \\ (v_j, v_i) &\in PA. \end{aligned}$$

An *authorization path* (or *au-path*) between v_1 and v_n is a sequence of vertices v_1, \dots, v_n such that $(v_i, v_{i+1}) \in E$, $i = 1, \dots, n - 1$. Hence, a user u can activate a role r if there is an au-path between u and r ; a role r is authorized for permission p if there is an au-path between r and p ; and a user u is authorized for permission p if there is an au-path between u and p . For simplicity, we introduce the following definition.

DEFINITION 1. *An entity $v \in U \cup R$ is RBAC96-authorized for $v' \in R \cup P$ if and only if there exists an*

¹The nature of a permission depends largely on the implementation details of a system, and some authors prefer to treat permissions as “uninterpreted symbols” [14].

au-path $v = v_1, v_2, \dots, v_n = v'$.²

2.3 ERBAC07 syntax

It has been observed that there are a number of situations where it is necessary to distinguish between role activation and permission usage [13]. In particular, it solves certain issues that arise when there is a role hierarchy and separation of duty constraints (as in RBAC₃). The ERBAC96 (extended RBAC96) model introduces a separate role *activation hierarchy*, a relation which is a superset of the (permission) *usage hierarchy* [13]. This means that it is necessary for a user to explicitly activate certain roles in order to obtain authorization for the permissions associated with those roles.

Chen and Crampton recently introduced ERBAC07 [5], which defines the activation and usage hierarchies by replacing the standard role hierarchy relation with a new relation $RH = R \times R \times \{a, u\}$. The activation hierarchy is denoted by $RH_a = \{(r, r') : (r, r', a) \in RH\}$, and the permission usage hierarchy is denoted by $RH_u = \{(r, r') : (r, r', u) \in RH\}$. We write \leq_a to denote the reflexive transitive closure of RH_a and \leq_u to denote the reflexive transitive closure of RH_u . In other words, the usage and activation hierarchies are modeled as two partial orderings on the set of roles R . ERBAC07 does not require that $RH_u \subseteq RH_a$; otherwise it is semantically equivalent to ERBAC96.

2.4 ERBAC07 semantics

We construct an acyclic, directed graph $G = (V, E)$, where $V = U \cup R \cup P$, and $E = UA \cup PA \cup RH_a \cup RH_u$. An *activation path* (or *a-path*) between v_1 and v_n is defined to be a sequence of vertices v_1, \dots, v_n such that $(v_1, v_2) \in UA$ and $(v_{i+1}, v_i) \in RH_a$ for $i = 2, \dots, n-1$. A *usage path* (or *u-path*) between v_1 and v_n is defined to be a sequence of vertices v_1, \dots, v_n such that $(v_{i+1}, v_i) \in RH_u$ ($i = 1, \dots, n-2$) and $(v_n, v_{n-1}) \in PA$. In ERBAC07:

- $v \in U$ may activate role $v' \in R$ if and only if there exists an a-path $v = v_1, v_2, \dots, v_n = v'$;
- $v \in R$ is authorized for permission $v' \in P$ if and only if there exists a u-path $v = v_1, v_2, \dots, v_n = v'$;
- $v \in U$ is authorized for permission $v' \in P$ if and only if there exists a path $v = v_1, v_2, \dots, v_i, \dots, v_n = v'$ such that $v_i \in R$ for some i , v_1, \dots, v_i is an a-path, and v_i, \dots, v_n is a u-path.

We say v_1, \dots, v_n is an au-path in ERBAC07 if v_1, \dots, v_n is either an a-path, or a u-path, or the concatenation of an a-path and an u-path.

DEFINITION 2. An entity $v \in U \cup R$ is ERBAC07-authorized for $v' \in R \cup P$ if and only if there exists an au-path $v = v_1, \dots, v_n = v'$.

3. SPATIO-TEMPORAL RBAC

We assume the existence of the usual RBAC96 entities: U , R , P , UA , PA , and RH ; we write V to denote $U \cup R \cup P$ and E to denote $UA \cup PA \cup RH$. We also assume the existence of a spatio-temporal domain \mathcal{D} : $d \in \mathcal{D}$ represents a point

²Note that $r \in R$ is RBAC96-authorized for $r' \in R$ means that r is senior to r' in the role hierarchy.

in space-time; $D \subseteq \mathcal{D}$ represents a collection of points in space-time.³

3.1 The standard model

The *standard spatio-temporal RBAC model* (or RBAC_{ST}⁼) augments the standard RBAC96 model with a function $\lambda : V \rightarrow 2^{\mathcal{D}}$. For $v \in V$, $\lambda(v) \subseteq \mathcal{D}$ denotes the set of points in space-time at which v is “enabled”. In particular,

- if $u \in U$, then $\lambda(u)$ denotes the set of points in space-time at which u may create a session;
- if $r \in R$, then $\lambda(r)$ denotes the set of points in space-time at which a role may be activated in a session;
- if $p \in P$, then $\lambda(p)$ denotes the set of points in space-time at which a permission may be granted.

Given a path v_1, \dots, v_n in the labeled graph $G = (V, E, \lambda)$, we write $\hat{\lambda}(v_1, \dots, v_n) \subseteq \mathcal{D}$ to denote $\bigcap_{i=1}^n \lambda(v_i)$. In other words, $\hat{\lambda}(v_1, \dots, v_n)$ is the set of points at which every vertex v_i is enabled. When the context is clear, we will write $\hat{\lambda}(v_1, v_n)$ for $\hat{\lambda}(v_1, \dots, v_n)$.

DEFINITION 3. An entity $v \in U \cup R$ is RBAC_{ST}⁼-authorized for $v' \in R \cup P$ at point $d \in \mathcal{D}$ if and only if there exists an au-path $v = v_1, v_2, \dots, v_n = v'$ and $d \in \hat{\lambda}(v, v')$.

The above definition naturally imposes the following constraint on λ in the RBAC_{ST}⁼ model.

CONSTRAINT 4. If $e = (v, v') \in E$, then $\lambda(v) \cap \lambda(v') \neq \emptyset$.

We now introduce a running example, which will be used to motivate the additional models that we define. Figure 1 illustrates six directed graphs for different user-role assignments and role hierarchies: vertices u and v represent users, and vertices r , r' and r'' represent roles.

Let us assume that we want to express the following spatio-temporal constraints:

- any user assigned explicitly to role r can only activate this role in spatio-temporal domain $D \subseteq \mathcal{D}$;
- any user assigned explicitly to role r' can activate role r from any point $d \in \mathcal{D}$;
- any user may activate role r'' from any point $d \in \mathcal{D}$.

For concreteness, r might be a clerical role and users occupying this role may only activate this role if they are in some particular part of the office building. In contrast, r' is a managerial role and a user occupying this role may activate the clerical role when she is sitting in her own office (or anywhere else); r'' is a general employee role and can be activated from anywhere in the office.

It is obvious that we can specify these requirements using RBAC_{ST}⁼ for the configurations shown in Figures 1(a) and 1(b). In particular, for Figure 1(a), we could define $\lambda(u) = D$ or $\lambda(r) = D$ or $\lambda(u) = \lambda(r) = D$. However, for Figure 1(b), we must define $\lambda(u) = D$, as user v , assigned to role r' , is allowed to activate role r at any $d \in \mathcal{D}$.

³In Section 6 we elaborate on possible representations of \mathcal{D} . For the purposes of the discussion in this section, it is sufficient to assume the existence of some abstract spatio-temporal domain.

Now consider the configuration in Figure 1(c), in which u is also assigned to role r'' . Since u is allowed to activate r'' at any $d \in \mathcal{D}$, we can not define $\lambda(u) = D$; nor can we set $\lambda(r) = D$, as v is allowed to activate r at any $d \in \mathcal{D}$. Hence, we require a spatio-temporal constraint on edge (u, r) . In other words, $\text{RBAC}_{\overline{ST}}$ is not sufficiently expressive for certain RBAC96 configurations and spatio-temporal requirements. For this reason, we now introduce a second model.

3.2 The strong model

The *strong spatio-temporal RBAC model* ($\text{RBAC}_{\overline{ST}}^+$) augments the $\text{RBAC}_{\overline{ST}}$ model with a function $\mu : E \rightarrow 2^{\mathcal{D}}$. For $e = (v, v') \in E$, $\mu(v, v')$ denotes the set of points in space-time at which the association between v and v' is enabled. In particular,

- if $(u, r) \in UA$, $\mu(u, r)$ denotes the set of points in space-time at which u is assigned to r ;
- if $(r, r') \in RH$, $\mu(r', r)$ denotes the set of points in space-time at which r' is senior to r ;
- if $(p, r) \in PA$, $\mu(r, p)$ denotes the set of points in space-time at which p is assigned to r .

Given a path v_1, \dots, v_n in the labeled graph $G = (V, E, \lambda, \mu)$, we write $\widehat{\mu}(v_1, \dots, v_n)$ to denote $\bigcap_{i=1}^{n-1} \mu(v_i, v_{i+1})$. Note that the semantics of $\text{RBAC}_{\overline{ST}}$ imply that an edge can only be enabled if both end points are enabled. Hence, $\widehat{\mu}(v, \dots, v')$ is the set of points at which every node and every edge in the path is enabled. When the context is clear, we will write $\widehat{\mu}(v_1, v_n)$ for $\widehat{\mu}(v_1, \dots, v_n)$.

We define the following constraint on μ .

CONSTRAINT 5. *If $(v, v') \in E$, then $\emptyset \subset \mu(v, v') \subseteq \lambda(v) \cap \lambda(v')$.*

DEFINITION 6. *An entity $v \in U \cup R$ is $\text{RBAC}_{\overline{ST}}^+$ -authorized for $v' \in R \cup P$ at point d if and only if there exists an au-path $v = v_1, v_2, \dots, v_n = v'$ and $d \in \widehat{\mu}(v, v')$.*

Note that $\text{RBAC}_{\overline{ST}}$ is a special case of $\text{RBAC}_{\overline{ST}}^+$ in which $\mu(v, v') = \lambda(v) \cap \lambda(v')$. In other words, we can express $\text{RBAC}_{\overline{ST}}$ using $\text{RBAC}_{\overline{ST}}^+$.

Consider Figures 1(c) and 1(d). We can define $\mu(u, r) = D$ to express our spatio-temporal requirements in $\text{RBAC}_{\overline{ST}}^+$.

However, $\text{RBAC}_{\overline{ST}}^+$ and $\text{RBAC}_{\overline{ST}}$ cannot be used to express these requirements given the configuration in Figure 1(e). In particular, we cannot define $\lambda(r) = D$ in $\text{RBAC}_{\overline{ST}}$ or $\mu(u, r) = D$ in $\text{RBAC}_{\overline{ST}}^+$ because this will only allow u to activate r'' at points $d \in D$. We now introduce a third model with weaker restrictions on valid authorization paths.

3.3 The weak model

Like $\text{RBAC}_{\overline{ST}}$, the *weak spatio-temporal RBAC model* (or $\text{RBAC}_{\overline{ST}}^-$) augments the standard RBAC96 model with a function $\lambda : V \rightarrow 2^{\mathcal{D}}$. In $\text{RBAC}_{\overline{ST}}^-$, the authorization semantics are different from those in $\text{RBAC}_{\overline{ST}}$.

DEFINITION 7. *An entity $v \in U \cup R$ is $\text{RBAC}_{\overline{ST}}^-$ -authorized for $v' \in R \cup P$ at point d if and only if there exists an au-path $v = v_1, v_2, \dots, v_n = v'$ and $d \in \lambda(v) \cap \lambda(v')$.*

In other words, an entity v is $\text{RBAC}_{\overline{ST}}^-$ -authorized for another entity v' if v is RBAC96-authorized for v' , and both

entities v and v' are enabled. There is no requirement that intermediate nodes on the path are enabled. These semantics appear to be closest to those defined in GTRBAC and the model of Ray and Toahchoodee. However, we would argue that $\text{RBAC}_{\overline{ST}}^-$ has the least intuitive semantics: why is it appropriate to ignore the enabling conditions on intermediate roles? There may be occasions when it is convenient to do so, as in Figure 1(e), but ignoring the intermediate roles is unlikely to be appropriate in many situations, and is inconsistent with the usual interpretation of inheritance in a role hierarchy. We would argue that the standard or strong models, in which enabling conditions are inherited up the hierarchy, are more closely aligned with standard RBAC semantics.

Consider Figure 1(e). Using the weak model, we can define $\lambda(r) = D$ to express our spatio-temporal requirements.

However, we cannot express our spatio-temporal requirements for the configuration shown in Figure 1(f) using any of the models we have defined. If we use $\text{RBAC}_{\overline{ST}}$, then we require that $\lambda(r) = D$ in order to restrict u 's activation of r . This, in turn means that v is unable to activate r from any point $d \notin D$. However, if we use $\text{RBAC}_{\overline{ST}}^-$ or $\text{RBAC}_{\overline{ST}}^+$, we must define $\lambda(u) = D$ or $\mu(u, r) = D$, which means that u is unable to activate r'' from any point $d \notin D$.

3.4 Trusted entities

We introduce the idea of *trusted entities*, which may be a user or a role; we write $T \subseteq U \cup R$ to denote the set of trusted entities. For an entity $t \in T$, the enabling constraints on nodes/edges in the authorization path from t are ignored.⁴

DEFINITION 8. *An entity $v \in U \cup R$ is $\text{RBAC}_{\overline{ST}}^-$ -authorized for $v' \in R \cup P$ at point $d \in \mathcal{D}$ if and only if there exists an au-path $v = v_1, \dots, v_j, \dots, v_n = v'$ such that $v_j \in T$ and $d \in \widehat{\lambda}(v, v_j)$, or there exists an au-path $v = v_1, v_2, \dots, v_n = v'$ and $d \in \widehat{\lambda}(v, v')$.*

DEFINITION 9. *An entity $v \in U \cup R$ is $\text{RBAC}_{\overline{ST}}^+$ -authorized for $v' \in R \cup P$ at point d if and only if there exists an au-path $v = v_1, \dots, v_j, \dots, v_n = v'$ such that $v_j \in T$ and $d \in \widehat{\mu}(v, v_j)$, or there exists an au-path $v = v_1, v_2, \dots, v_n = v'$ and $d \in \widehat{\mu}(v, v')$.*

DEFINITION 10. *An entity $v \in U \cup R$ is $\text{RBAC}_{\overline{ST}}^-$ -authorized for $v' \in R \cup P$ at point $d \in \mathcal{D}$ if and only if there exists an au-path $v = v_1, \dots, v_j, \dots, v_n = v'$ such that $v_j \in T$ and $d \in \lambda(v) \cap \lambda(v_j)$, or there exists an au-path $v = v_1, v_2, \dots, v_n = v'$ and $d \in \lambda(v) \cap \lambda(v')$.*

Consider Figure 1(f). In order to express our spatio-temporal requirements, we use $\text{RBAC}_{\overline{ST}}^-$ and define r' (or v) to be a trusted entity and $\lambda(r) = D$. Clearly, user v can activate roles r and r'' from any point because there exists an au-path v, r', r (and the fact that $\lambda(r) = D$ is ignored).

⁴The interpretation of a privileged entity is similar to that of a privileged method in the Java runtime environment (JRE). The stackwalking algorithm, which is used to perform access control in the JRE, normally examines the permissions of every method on the stack. Access is only granted if every method on the stack has the requested permission. However, the stackwalk terminates prematurely if a privileged method is encountered on the stack (thereby ignoring any methods lower down the stack that may not have the requested permission).

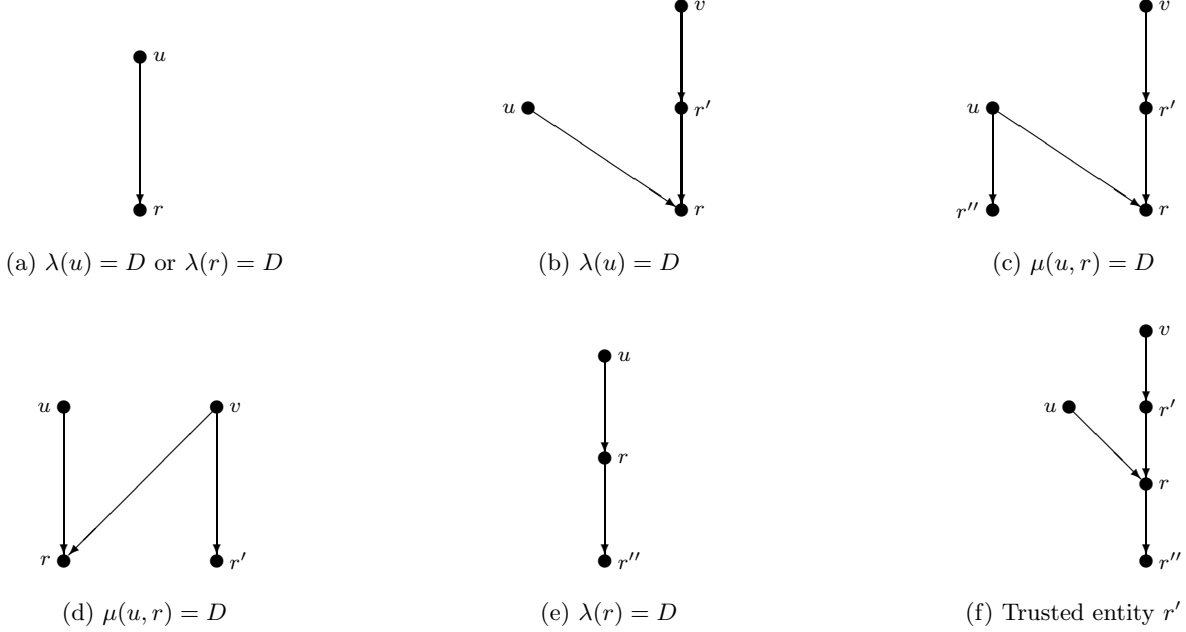


Figure 1: Configurations of RBAC96 model

3.5 A note on RBAC96-style syntax

We currently use the functions λ and μ to define the syntax of our spatio-temporal RBAC models, and a graph-based formalism to define the semantics of these models. In this section, we briefly note that we can use RBAC96-style syntax to encode RBAC_{ST}^+ . (It follows that RBAC_{ST}^- and $\text{RBAC}_{ST}^{\bar{-}}$ syntax can also be adjusted in the same way.)

The familiar sets and relations from the RBAC96 model – U , R , P , UA , RH and PA – are adjusted to include an extra entry, corresponding to the set of points for which the entity or entity relationship is enabled. The set of users U , for example, is replaced by $U_{ST} \subseteq U \times 2^{\mathcal{D}}$; $(u, D) \in U_{ST}$ means that u is enabled for all points $d \in D$. The set of user-role assignments UA , for example, is replaced by $UA_{ST} \subseteq U \times R \times 2^{\mathcal{D}}$; $(u, r, D) \in UA_{ST}$ means that the assignment of user u to role r is enabled for all points $d \in D$. In $\text{RBAC}_{ST}^{\bar{-}}$, for example, a user u may activate a role r at point d if there exist roles $r' = r_1, r_2, \dots, r_n = r$, such that $(r_{i+1}, r_i) \in RH$, $(u, r') \in UA$, $(u, D_u) \in U_{ST}$, $(r_i, D_i) \in R_{ST}$, $i = 1, \dots, n$, and $d \in D_u \cap D_1 \cap \dots \cap D_n$.

3.6 Integration with ANSI-RBAC

The core and hierarchical components of ANSI-RBAC standard are defined by a set of basic element sets U , S , R and P , a set of relations UA , RH and PA , and a set of mapping functions, shown in the top part of Table 1.

The table demonstrates that it is easy to re-define the ANSI-RBAC functions in the context of $\text{RBAC}_{ST}^{\bar{-}}$ and RBAC_{ST}^+ .⁵ The function *session_users*, also defined by the ANSI-RBAC standard, returns the user associated with a session, and is the same for all three models. Each function, when defined for $\text{RBAC}_{ST}^{\bar{-}}$ and RBAC_{ST}^+ , includes a param-

⁵For brevity, we omit RBAC_{ST}^- and trusted entities from this discussion.

eter $d \in \mathcal{D}$. For simplicity we use our original syntax, rather than the RBAC96-style syntax.

The ANSI-RBAC standard defines functions *avail_session_perms* and *session_roles*, shown in the first section of Table 1. These function only apply to the core component; no analogous function is defined for the hierarchical component, which is a curious omission. Instead, the standard defines two additional functions – *authorized_users* and *authorized_permissions*, each of which take a role as a parameter.

We propose new definitions for the functions *session_roles* and *avail_session_perms* for the hierarchical component. These are shown in the second section of Table 1. Note that in core RBAC, *assigned_permissions*(r) = *authorized_permissions*(r) for all r . Hence, this definition is consistent with that given in the ANSI-RBAC standard.

4. SPATIO-TEMPORAL ERBAC

In this section we extend the spatio-temporal model we have developed for RBAC96 to include the features defined in ERBAC07.

4.1 The standard model

The *standard spatio-temporal ERBAC model* (or $\text{ERBAC}_{ST}^{\bar{-}}$) combines the features of $\text{RBAC}_{ST}^{\bar{-}}$ and ERBAC07. In other words, we extend the directed labeled graph (V, E, λ) , where $E = UA \cup RH_a \cup RH_u \cup PA$.

DEFINITION 11. In $\text{ERBAC}_{ST}^{\bar{-}}$:

- a user $v \in U$ may activate role $v' \in R$ at point $d \in \mathcal{D}$ if and only if there exists an a -path $v = v_1, v_2, \dots, v_n = v'$ and $d \in \hat{\lambda}(v, v')$;
- a role $v \in R$ is authorized for permission $v' \in P$ at

ANSI-RBAC
$assigned_users(r) = \{u \in U : (u, r) \in UA\}$ $assigned_permissions(r) = \{p \in P : (p, r) \in PA\}$ $session_users(s) = u$ $session_roles(s) \subseteq \{r \in R : (session_users(s), r) \in UA\}$ $authorized_users(r) = \{u \in U : r \leq r', (u, r') \in UA\}$ $authorized_permissions(r) = \{p \in P : r \geq r', (p, r') \in PA\}$ $avail_session_perms(s) = \bigcup_{r \in session_roles(s)} assigned_permissions(r)$
Proposed extensions for hierarchical ANSI-RBAC
$session_roles(s) \subseteq \{r \in R : r \leq r', (session_users(s), r') \in UA\}$ $avail_session_perms(s) = \bigcup_{r \in session_roles(s)} authorized_permissions(r)$
RBAC $_{ST}^-$
$assigned_users(r, d) = \{u \in U : (u, r) \in UA, d \in \lambda(u) \cap \lambda(r)\}$ $assigned_permissions(r, d) = \{p \in P : (p, r) \in PA, d \in \lambda(p) \cap \lambda(r)\}$ $session_roles(s, d) \subseteq \{r \in R : r \leq r', (session_users(s), r') \in UA, d \in \widehat{\lambda}(r', r)\}$ $authorized_users(r, d) = \{u \in U : r \leq r', (u, r') \in UA, d \in \lambda(u) \cap \widehat{\lambda}(r', r)\}$ $authorized_permissions(r, d) = \{p \in P : r \geq r', (p, r') \in PA, d \in \lambda(p) \cap \widehat{\lambda}(r, r')\}$ $avail_session_perms(s, d) = \bigcup_{r \in session_roles(s, d)} authorized_permissions(r, d)$
RBAC $_{ST}^+$
$assigned_users(r, d) = \{u \in U : (u, r) \in UA, d \in \mu(u, r)\}$ $assigned_permissions(r, d) = \{p \in P : (p, r) \in PA, d \in \mu(p, r)\}$ $session_roles(s, d) \subseteq \{r \in R : r \leq r', (session_users(s), r') \in UA, d \in \widehat{\mu}(session_users(s), r)\}$ $authorized_users(r, d) = \{u \in U : r \leq r', (u, r') \in UA, d \in \widehat{\mu}(u, r)\}$ $authorized_permissions(r, d) = \{p \in P : r \geq r', (p, r') \in PA, d \in \widehat{\mu}(r, p)\}$ $avail_session_perms(s, d) = \bigcup_{r \in session_roles(s, d)} authorized_permissions(r, d)$

Table 1: ANSI-RBAC mapping functions

point $d \in \mathcal{D}$ if and only if there exists a u -path $v = v_1, v_2, \dots, v_n = v'$ and $d \in \widehat{\lambda}(v, v')$;

- a user $v \in U$ is authorized for permission $v' \in P$ at point $d \in \mathcal{D}$ if and only if there exists a path $v = v_1, v_2, \dots, v_i, \dots, v_n = v'$ such that $v_i \in R$ for some i , v_1, \dots, v_i is an a -path, v_i, \dots, v_n is a u -path, and $d \in \widehat{\lambda}(v, v')$.

4.2 The strong model

The *strong spatio-temporal ERBAC model* (or ERBAC $_{ST}^+$) combines the features of RBAC $_{ST}^+$ and ERBAC07. In other words, we have the extended directed labeled graph (V, E, λ, μ) , where $E = UA \cup RH_a \cup RH_u \cup PA$.

DEFINITION 12. In ERBAC $_{ST}^+$:

- a user $v \in U$ may activate role $v' \in R$ at point $d \in \mathcal{D}$ if and only if there exists an a -path $v = v_1, v_2, \dots, v_n = v'$, and $d \in \widehat{\mu}(v, v')$;
- a role $v \in R$ is authorized for permission $v' \in P$ at point $d \in \mathcal{D}$ if and only if there exists a u -path $v = v_1, v_2, \dots, v_n = v'$, and $d \in \widehat{\mu}(v, v')$;
- a user $v \in U$ is authorized for permission $v' \in P$ at point $d \in \mathcal{D}$ if and only if there exists a path $v = v_1, v_2, \dots, v_i, \dots, v_n = v'$ such that $v_i \in R$ for some i , v_1, \dots, v_i is an a -path, v_i, \dots, v_n is a u -path, and $d \in \widehat{\mu}(v, v')$.

4.3 The weak model

The *weak spatio-temporal ERBAC model* (or ERBAC $_{ST}^-$) combines the features of RBAC $_{ST}^-$ and ERBAC07. Like ERBAC $_{ST}^-$, we have the extended directed labeled graph (V, E, λ) , where $E = UA \cup RH_a \cup RH_u \cup PA$.

DEFINITION 13. In ERBAC $_{ST}^-$:

- a user $v \in U$ may activate role $v' \in R$ at point $d \in \mathcal{D}$ if and only if there exists an a -path $v = v_1, v_2, \dots, v_n = v'$, and $d \in \lambda(v) \cap \lambda(v')$;
- a role $v \in R$ is authorized for permission $v' \in P$ at point $d \in \mathcal{D}$ if and only if there exists a u -path $v = v_1, v_2, \dots, v_n = v'$, and $d \in \lambda(v) \cap \lambda(v')$;
- a user $v \in U$ is authorized for permission $v' \in P$ at point $d \in \mathcal{D}$ if and only if there exists a path $v = v_1, v_2, \dots, v_i, \dots, v_n = v'$ such that $v_i \in R$ for some i , v_1, \dots, v_i is an a -path, v_i, \dots, v_n is a u -path, and $d \in \lambda(v) \cap \lambda(v_i) \cap \lambda(v')$.

5. PRACTICAL CONSIDERATIONS IN SPATIO-TEMPORAL RBAC

5.1 Is the use of hierarchies realistic?

The examples in Figure 1 illustrate that the presence of a role hierarchy significantly complicates the specification

of spatio-temporal constraints. We argued in Section 3 that there were at least three different models that could be used; even then, it was necessary to introduce the notion of trusted entities for certain scenarios. This suggests that there are many possible interpretations of spatio-temporal restrictions in the presence of a role hierarchy. Choosing the appropriate model may well be difficult, and encoding the desired enterprise security policies within such a model is also likely to be non-trivial. In the next two sections, we consider four simple strategies that might be used to mitigate these difficulties.

5.1.1 Flat spatio-temporal RBAC

In practical applications, it might well be preferable to assume that the set of roles is unordered, as in core ANSI-RBAC standard or flat RBAC96 (RBAC₀). This means, of course, that the number of user- and permission-role assignments will increase (because such assignments are often implicitly generated by assignments to other roles in the presence of a role hierarchy). However, it does mean that flat RBAC_{ST}⁺ is sufficient for specifying all spatio-temporal constraints.

Consider the spatio-temporal requirements for the configuration of RBAC in Figure 1(f). We transform the RBAC96 configuration to flat RBAC as follows: $U = \{u, v\}$, $R = \{r, r', r''\}$ and $UA = \{(v, r'), (v, r), (v, r''), (u, r), (u, r'')\}$. We only need to define $\mu(u, r) = D$; all other nodes and edges are enabled for any $d \in \mathcal{D}$. It is obvious that we can easily encode the spatio-temporal requirements for other configurations of RBAC in Figure 1 using flat RBAC_{ST}⁺.

5.1.2 Eliminate enabling restrictions on roles

An alternative approach is to eliminate enabling restrictions on roles: that is, set $\lambda(r) = \mathcal{D}$ for all $r \in R$. In other words, all roles are enabled at all points in the spatio-temporal domain. This approach is completely contrary to existing approaches, in which roles are usually the only entities for which such enabling conditions are defined. In this approach, restrictions are imposed at the outer nodes and edges of the RBAC graph. As such, it extends the approach advocated in the previous section by including a role hierarchy.

An example that is often quoted in the temporal RBAC literature is that of a **night-doctor** role, which should only be enabled during the night shift hours. We would argue that instead of imposing the enabling condition on the **night-doctor** role, we should impose the condition on any assignment of that role to a user. This does not preclude the same user from also being assigned to the **day-doctor** role (which would have a different enabling condition on the user-role assignment).

It would not be difficult to implement this kind of approach. Let us assume that we have a **night-doctor** role, which should only be activated during the night shift. Then, whenever a user is assigned to this role, an enabling condition is automatically generated for that user-role assignment. (If the intersection of the user's enabling condition and this condition is empty, then the assignment fails.)

Let us now consider the impact of setting $\lambda(r) = \mathcal{D}$ for all r . Then, in RBAC_{ST}⁼, a user u may activate a role r at point d if there is an au-path $u, r_1, \dots, r_n = r$ and $d \in \lambda(u)$, a role r is authorized for permission p at point d if there is an au-path $r = r_1, \dots, r_n, p$ and $d \in \lambda(p)$, and user u is authorized for permission p at point d if there is an au-

path u, r_1, \dots, r_n, p and $d \in \lambda(u) \cap \lambda(p)$; in RBAC_{ST}⁺, a user u may activate a role r at point d if there is an au-path $u, r_1, \dots, r_n = r$ and $d \in \mu(u, r_1)$, a role r is authorized for permission p at point d if there is an au-path $r = r_1, \dots, r_n, p$ and $d \in \mu(r_n, p)$, and user u is authorized for permission p at point d if there is an au-path u, r_1, \dots, r_n, p and $d \in \mu(u, r_1) \cap \mu(r_n, p)$.

5.2 Partial transitive closure

Let us now assume, however, that there *is* a requirement for a role hierarchy and for having enabling conditions on the roles. We note that checking whether a user may activate a role or is granted a permission may be a relatively complex operation when there are spatio-temporal constraints and a role hierarchy. This is because there may be multiple paths between two roles in a role hierarchy and because we need to check whether the point at which the access request was made belongs to each of the enabling conditions. Hence, we suggest that in practical implementations, it might be useful to pre-compute the transitive closure of (part of) the RBAC96 graph.

One possibility is to construct RH^* , the transitive closure of RH , and assign $D \subseteq \mathcal{D}$ to $(r, r') \in RH^*$.

5.2.1 RBAC_{ST}⁼

In RBAC_{ST}⁼, for example, this value would be the union of the individual $\hat{\lambda}$ values for each path between r and r' . That is, given $r, r' \in R$, let $\pi(r, r')$ denote the set of paths between r and r' , and for $p \in \pi(r, r')$, let $\hat{\lambda}(p, r, r')$ denote $\hat{\lambda}(r, r')$ for path p . We define $\hat{\lambda}^* : RH^* \rightarrow 2^{\mathcal{D}}$, where

$$\hat{\lambda}^*(r, r') = \bigcup_{p \in \pi(r, r')} \hat{\lambda}(p, r, r')$$

Suppose, for example, that $r_4 < r_2 < r_1$ and $r_4 < r_3 < r_1$ and that r_2 and r_3 are incomparable. Suppose also that $\lambda(r_i) = D_i$. Then

$$\begin{aligned} \hat{\lambda}^*(r_1, r_4) &= (D_1 \cap D_2 \cap D_4) \cup (D_1 \cap D_3 \cap D_4) \\ &= D_1 \cap D_4 \cap (D_2 \cup D_3). \end{aligned}$$

We represent the partial transitive closure of RBAC_{ST}⁼ as a tuple $(V, E^*, \lambda, \hat{\lambda}^*)$, where $E^* = UA \cup RH^* \cup PA$, $\lambda : V \rightarrow 2^{\mathcal{D}}$ and $\hat{\lambda}^* : RH^* \rightarrow 2^{\mathcal{D}}$. Given $G^* = (V, E^*, \lambda, \hat{\lambda}^*)$, a request by u to exercise a permission p at point d is granted if u has activated a role r_1 at d and there exists (r_1, r_n) and (r_n, p) in E^* such that $d \in \hat{\lambda}^*(r_1, r_n) \cap \lambda(p)$.

5.2.2 RBAC_{ST}⁺

Similarly, in RBAC_{ST}⁺, for $p \in \pi(r, r')$, let $\hat{\mu}(p, r, r')$ denote $\hat{\mu}(r, r')$ for path p . We define $\hat{\mu}^* : RH^* \rightarrow 2^{\mathcal{D}}$, where

$$\hat{\mu}^*(r, r') = \bigcup_{p \in \pi(r, r')} \hat{\mu}(p, r, r')$$

We represent the partial transitive closure of RBAC_{ST}⁺ as a tuple $(V, E^*, \lambda, \mu, \hat{\mu}^*)$. Given $G^* = (V, E^*, \lambda, \mu, \hat{\mu}^*)$, a request by u to exercise a permission p at point d is granted if u has activated a role r_1 at d and there exists (r_1, r_n) and (r_n, p) in E^* such that $d \in \hat{\mu}^*(r_1, r_n) \cap \mu(r_n, p)$.

5.2.3 Spatio-temporal ERBAC07

For the models based on ERBAC07, we compute RH_a^* , the transitive closure of RH_a , and RH_u^* , the transitive closure

of RH_u , and define functions $\hat{\lambda}_a^*$, $\hat{\lambda}_u^*$, $\hat{\mu}_a^*$ and $\hat{\mu}_u^*$. We omit further details.

5.3 Full transitive closure

We now briefly consider the full transitive closure of G , $G^* = (V, E^*)$, where $E^* = (UA \cup RH \cup PA)^*$. In $\text{RBAC}_{\overline{ST}}$, given $v, v' \in V$, let $\pi(v, v')$ denote the set of paths between v and v' , and for $p \in \pi(v, v')$, let $\hat{\lambda}(p, v, v')$ denote $\hat{\lambda}(v, v')$ for path p . We define $\hat{\lambda}^* : E^* \rightarrow 2^{\mathcal{D}}$, where

$$\hat{\lambda}^*(v, v') = \bigcup_{p \in \pi(v, v')} \hat{\lambda}(p, v, v')$$

We represent the full transitive closure of $\text{RBAC}_{\overline{ST}}$ as a tuple $(V, E^*, \lambda, \hat{\lambda}^*)$. Given $G^* = (V, E^*, \lambda, \hat{\lambda}^*)$, a request by u to exercise permission p at point d is granted if there exists (u, p) in E^* such that $d \in \hat{\lambda}^*(u, p)$.

The full transitive closure of $\text{RBAC}_{\overline{ST}}^+$, $\text{ERBAC}_{\overline{ST}}$ and $\text{ERBAC}_{\overline{ST}}^+$ are similar, and are omitted. Computing the full transitive closure will only be practical for relatively small numbers of users and permissions, so it is likely that computing the partial transitive closure will be more useful in practice.

5.4 Concluding remarks

We have developed three spatio-temporal RBAC models and introduced the notion of trusted entities to specify the spatio-temporal requirements in different configurations of RBAC. The need for different models arises because once enabling conditions are imposed on roles, there are a number of different choices for the semantics of authorization. In practice, it is complicated and error-prone to specify comprehensive spatio-temporal requirements in hierarchical RBAC model. Therefore, we would argue that, in many practical situations, the most appropriate approach is to use flat $\text{RBAC}_{\overline{ST}}^+$ to specify spatio-temporal requirements.

However, when there are very large numbers of user and permissions, it may well be appropriate to use role hierarchies, thereby avoiding large numbers of user- and permission-role assignments. In this case, it is appropriate to set $\lambda(r) = \mathcal{D}$ for all $r \in R$, and specify enabling conditions on restrictions on outer nodes and edges, such as users and user-role assignments, of the RBAC graph. We should perhaps note that the underlying ‘‘philosophy’’ of RBAC is to use roles to reduce the burden of administration, and that our suggestion of applying enabling constraints to users and user-role assignment is inconsistent with this basic tenet. As we have seen, however, many situations may require constraints on users and user-role assignment, rather than roles. This suggests that incorporating spatio-temporal constraints within RBAC is likely to require some trade-off between the complexity of policies that can be supported and the complexity of constraint specification and administration.

In addition, when there are requirements for role hierarchy and enabling conditions on the roles, we suggest that it is appropriate to pre-compute the transitive closure of the role hierarchy to avoid complex computations when checking access requests. On the other hand, it is unlikely that it is useful to pre-compute the full transitive closure of $\text{RBAC}_{\overline{ST}}$ graph in many practical systems, because the size of E^* will be very large. However, deciding access requests can be performed far more quickly than in the other three approaches.

6. SPATIO-TEMPORAL DOMAINS

Much of the work in extending RBAC to include spatial and temporal restrictions on entities and entity relationships has spent a considerable amount of time on how these restrictions might be specified. The authors of GTRBAC, for example, define a syntax for temporal restrictions using the notion of *calendars* [9]. Although we believe that it is of much greater importance to understand the interaction between RBAC inheritance and such restrictions, we now briefly consider how sets of points within a spatio-temporal domain might be specified.

Broadly speaking, there are two possibilities: *concrete* and *symbolic* domains. A concrete domain makes use of actual points in space-time, whereas a symbolic domain uses labels as synonyms for sets of points in an associated concrete domain. We consider spatial and temporal domains separately. A single spatio-temporal domain \mathcal{D} can be treated as a pair $(\mathcal{S}, \mathcal{T})$, where \mathcal{S} is a spatial domain and \mathcal{T} is a temporal domain.

6.1 Representing location

A concrete spatial domain is defined by a co-ordinate system: we could use standard Euclidean space or we may use spherical or cylindrical co-ordinate systems, for example. The system chosen will be entirely dependent on the method by which user location is determined. For ease of exposition, we will define the concrete spatial domain to be $\mathcal{S} = \{(x, y) : x, y \in \mathbb{Z}\}$. In other words, points in space are defined by two integer co-ordinates.

An *atomic location* is defined to be a rectangle, which is defined by the co-ordinates of its lower-left and upper-right corners.⁶ That is, a rectangle is a pair $[l, r]$, where $l, r \in \mathcal{S}$. A *location* is the union of one or more disjoint atomic locations: clearly, the set of locations is a subset of $2^{\mathcal{S}}$ and λ maps an entity to a location.

Having defined a concrete spatial domain, we may define a symbolic spatial domain, in which locations are associated with labels. Symbolic locations may be defined to be the union of other symbolic locations; these symbolic locations may overlap. Having defined a set of symbolic locations, we must define a mapping from the set of symbolic locations to concrete locations. We may also use λ to map entities to symbolic locations, and then map the symbolic location to a concrete location.

Let $s \in \mathcal{S}$ be a point in the concrete spatial domain, and let $L \subseteq \mathcal{S}$ be a concrete location. We write $s \in L$ if s belongs to one of the atomic locations contained in L . If L is a union of symbolic locations, we write $s \in L$ to denote that s belongs to at least one of the symbolic locations contained in L .

6.2 Representing time

We assume the existence of a clock, whose ticks are indexed by the natural numbers \mathbb{N} .⁷ An *atomic interval* in

⁶Of course, we could define a location to be a circular region in the concrete spatial domain, by defining the center $c \in \mathcal{S}$ and radius $r \in \mathbb{Z}$ of the circle. Again, the definition of location will be determined by the method used to identify the position of a user.

⁷It should be noted that representing time will be more complex than this for many applications; typically a local time is relative to a location and a time of year. Our representation of time, as for location, is merely illustrative.

the concrete temporal domain $\mathcal{T} = \mathbb{N}$, is defined by a start point $t_1 \in \mathcal{T}$ and an end point $t_2 \in \mathcal{T}$, and written as $[t_1, t_2]$. An *interval* is defined to be the union of one or more disjoint atomic intervals; λ maps an entity to an interval.

We may also define a symbolic temporal domain, in which intervals are associated with labels. We could, for example, define the symbolic intervals `21:August:2007`, `Mondays:2007`, `WorkingHours` etc. We may use λ to map entities to symbolic intervals.

Let $t \in \mathcal{T}$ be a point in the concrete temporal domain, and let $I \subseteq \mathcal{T}$ be a concrete interval. We write $t \in I$ if t belongs to one of the atomic intervals contained in I . If I is the union of symbolic intervals, we write $t \in I$ to denote that t belongs to at least one of the symbolic intervals contained in I .

6.3 Example

In this section we present examples to illustrate the applications of spatial RBAC_{ST}^- , temporal RBAC_{ST}^- and spatio-temporal ERBAC_{ST}^+ in practical environment.

6.3.1 Spatial RBAC_{ST}^-

Figure 2 illustrates some of the ideas that we have introduced in this paper. Figure 2(a) lists a number of RBAC entities associated with a computer science department at a university. Figure 2(b) illustrates the relationships between these entities. A user u_2 , who is assigned to role r_1 , is allowed to activate roles r_2, r_3, r_4 in any session. In RBAC96 , u_2 is authorized to invoke permissions p_1, p_2, p_3, p_4 since any permission can be reached by u_2 via a path in the graph.

In order to define spatial constraints for this example, we describe the layout of a floor in the computer building, as shown in Figure 2(c). Figure 2(d) defines enabling constraints for the RBAC entities in Figure 2(a). Note that all roles are enabled everywhere within the computer building, as suggested in Section 5.1.2. For example, permission to access the ACM and IEEE libraries (p_2) is only allowed if the requester is in the seminar room (SR), Alice’s office (AO), or Bob’s office (BO). In Diane’s office, for example, permissions p_2 and p_3 are not enabled; however, Diane is allowed to activate r_3 (Admin staff), thereby enabling her to view staff profile.

6.3.2 Temporal RBAC_{ST}^-

Let us consider the graphical formulation of RBAC96 policies for the computer building shown in Figure 2(a) and 2(b). Let us assume that Figure 2(d) represents symbolic temporal domains for all entities of RBAC96 in the example of computer building. Then at a particular point of time 14:00, the permission p_3 is not enabled. All other entities are enabled, and related edges exist at time 14:00. For example, Alice is allowed to activate role r_2 to use the permission p_1 that is inherited from role r_4 at time instant 14:00.

6.3.3 Spatio-temporal ERBAC_{ST}^+

Consider the activation and usage hierarchies of ERBAC07 shown in Figures 3(a) and 3(b), respectively, and the user-role assignment and the permission-role assignment are as same as the configurations in Figure 2(b). For example, user u_2 is authorized to activate role r_1 , but is not thereby authorized for permission p_4 which is not inherited by r_1 in the permission usage hierarchy. Let us assume that Figure 2(a) represents ERBAC07 policies in the computer

building.

Figures 2(d) and 3(c) represent the spatio-temporal enabling conditions for RBAC entities and relations. Note that a user must explicitly activate the Admin staff role in order to use the permissions associated with this role. Note also that the specification of spatio-temporal domains on edges observes the consistency constraint between nodes and edges. At a particular spatio-temporal point (Alice’s office, 13:30), Alice can not activate the role (Academic staff), because Alice is not assigned academic staff role at point (AO,13:30) although both user (Alice) and role (Academic staff) are enabled at point (AO,13:30). On the other hand, at point (Diane’s office, 14:00), Bob can activate the role (Admin staff) to use the permission (View staff profile).

7. RELATED WORK

In this section we examine the GTRBAC model [9] and the spatio-temporal RBAC model of Ray and Toahchoodee [12] in more detail, and review other related work on context-based access control. We explain why we believe that our model is more attractive than related work according to several criteria: well-defined authorization semantics, syntactic completeness (constraints on all RBAC entities and relations), consistency (absence of conflicts, resolution of conflicts), and syntactic simplicity (number of predicates or functions).

7.1 GTRBAC

The temporal-RBAC model (TRBAC) introduces temporal constraints which limit the time during which a role is enabled and activated [3]. Generalized TRBAC (GTRBAC) is an extension of TRBAC that applies temporal constraints to the assignment of users and permissions to roles [9]. GTRBAC does not consider temporal constraints on users (sessions), permissions and role hierarchical relationships. Moreover, GTRBAC, unlike our models, does not impose any consistency constraints on the user- and permission-role assignments and role-role relationships.

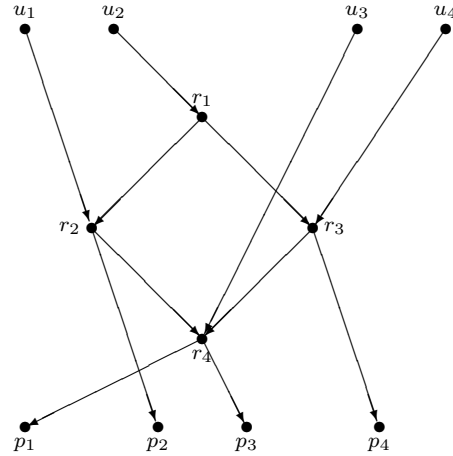
GTRBAC defines a “hybrid” role hierarchy that contains three different types of role hierarchy relationships: role-activation hierarchy \leq_a , permission-usage hierarchy \leq_u and permission-activation hierarchy \leq . However, the permission-activation hierarchy is redundant and can be defined in terms of other two hierarchies, that is $x \leq y$ if and only if $x \leq_a y$ and $x \leq_u y$. GTRBAC further sub-divide hierarchies into “weakly” and “strongly” restricted; the authorization semantics for these hierarchies differ. The weakly restricted semantics for permission usage [9, Table 7], are defined by

$$\begin{aligned} \text{can_be_acquired}(p, x, t) \leftarrow & \forall p, (x \geq_u y) \wedge \\ & \text{enabled}(x, t) \wedge \\ & \text{can_be_acquired}(p, y, t). \quad (1) \end{aligned}$$

The intuition seems to be that if x is enabled, $x \geq_u y$ and y can acquire permission p , then x can acquire permission p . To quote Joshi *et al.*: “The weakly restricted hierarchies allow inheritance or activation semantics in the nonoverlapping intervals. . . only role x needs to be enabled at time t for the [usage] inheritance semantics to apply”.

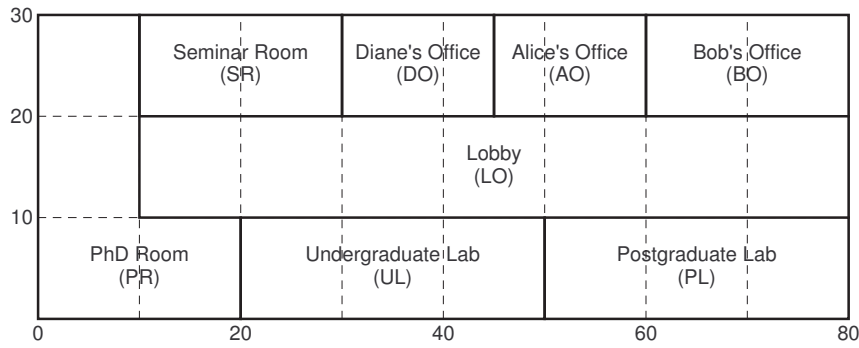
However, there are a number of problems with this definition. The predicate `can_be_acquired` is defined recursively, but there is no base case; in particular, replacing x

u_1	Alice
u_2	Bob
u_3	Chris
u_4	Diane
r_1	Head of department
r_2	Academic staff
r_3	Admin staff
r_4	Student
p_1	Access resources via Metalib
p_2	Access ACM and IEEE library
p_3	Listen to presentations
p_4	View staff profile



(a) RBAC entities

(b) Graphical representation of RBAC relations



(c) Spatial domain

Entity	Spatial domain		Temporal domain
	Symbolic	Concrete	Symbolic
u_1	CB	$[(0,0),(80,30)]$	09:00-17:59
u_2	CB	$[(0,0),(80,30)]$	09:00-17:59
u_3	CB	$[(0,0),(80,30)]$	Always
u_4	CB	$[(0,0),(80,30)]$	Always
r_1	CB	$[(0,0),(80,30)]$	Always
r_2	CB	$[(0,0),(80,30)]$	Always
r_3	CB	$[(0,0),(80,30)]$	Always
r_4	CB	$[(0,0),(80,30)]$	Always
p_1	CB	$[(0,0),(80,30)]$	Always
p_2	$SR \cup AO \cup BO$	$[(10,20),(30,30)] \cup [(45,20),(80,30)]$	09:00-17:59
p_3	SR	$[(10,20),(30,30)]$	12:00-13:00
p_4	DO	$[(30,20),(45,30)]$	Always

(d) Spatial-temporal enabling conditions

Figure 2: Spatio-temporal RBAC example

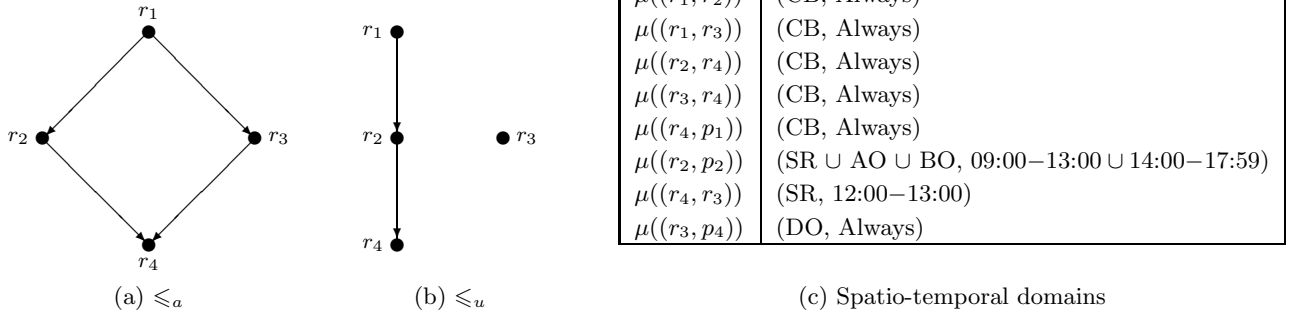


Figure 3: A graphical formulation of ERBAC⁺_{ST} policies in the computing building

with y (which is legitimate, since $y \geq y$) in the rule above means we have a circular definition. Presumably the base case is that $(p, y) \in PA$, but the presence of the parameter t in `can_be_acquired` suggests that there may be an enabling condition on this assignment. Similar problems exist for weakly restricted semantics for role activation, and for strongly restricted semantics for permission acquisition and role activation.

Without a base case, it is impossible to determine the intended meaning of weakly and strongly restricted hierarchies. Moreover, it seems that any enabling conditions on roles between x and y are ignored. This makes a direct comparison between our models and GTRBAC impossible. The strongly restricted semantics require x and y to be enabled, which suggests that strongly restricted semantics in GTRBAC are (intended to be) somewhat similar to RBAC⁻_{ST}.

7.2 Spatio-temporal RBAC

Ray and Toahchoodee developed a spatio-temporal RBAC model [12] that is strongly influenced by GTRBAC. Indeed, the main novelty of their approach is to introduce spatial and temporal constraints on all the components of RBAC. They also consider the consistency of the constraints on user-role and permission-role assignments.

Like ERBAC07, they introduce a role activation hierarchy \leq_a and a permission usage hierarchy \leq_u . They also define temporal constraints, location constraints, and temporal and location constraints on these two role hierarchies. Let us consider the representative example of “time location restricted permission inheritance hierarchy” [12, Definition 13], where

$$\text{PermRoleAcquire}(p, x, d, l) \leftarrow \forall p, (x \geq_u y) \wedge \text{PermRoleAcquire}(p, y, d, l). \quad (2)$$

Here, d represents a set of time points and l a set of points in space. Again, it is not clear what the base case is, and intermediate roles between x and y are ignored.

In addition, this definition may give rise to conflicts within the specification of enabling conditions. If `PermRoleAcquire`(p, r, d, l) holds then r and p are enabled at all points within d and l [12, Section 4.5]. Now let us

assume that

- `RoleEnableLoc`(x) = l' (x is enabled at l') and `RoleEableDur`(x) = d' (x is enabled during d'),
- `PermRoleAcquire`(p, y, d, l) holds and $x \geq_u y$,
- $d' \subset d$ and $l' \subset l$.

Then we have `PermRoleAcquire`(p, x, d, l), by (2). This implies that x is enabled at $l \supset l'$ and $d \supset d'$, which contradicts the enabling conditions defined on x . Similar conflicts exist for weakly temporal and location restricted permission acquisition.

7.3 Summary of other work

Work has been done on spatial constraints in the context of mandatory access control (MAC) [11], discretionary access control (DAC) [2] and RBAC models [4, 8]. This work has either studied spatial constraints in traditional access control models [2, 11], rather than RBAC, or proposed a limited spatially constrained RBAC [8]. GEO-RBAC [4] introduces a comprehensive spatial RBAC model for specifying spatial constraints on roles and treat locations as objects in RBAC model. They also introduce GEO-HRBAC model that defines the role hierarchy based on the containment of locations. Compared with our models, we believe that GEO-HRBAC is too application-dependent, and focuses on controlling access on different locations.

There has also been research on more general context information to achieve fined-grained role-based access control. Team-based access control (TMAC) [7, 16] approach extends RBAC with the notion of team and context-based permission activation. Covington *et al* [6] introduce the concept of environment roles in RBAC which are activated based on the values of environmental conditions. Strembeck *et al* [15] introduce the concept of context constraints in RBAC which is used to restrict usage of permissions through considering environmental factors in access control decision. Although all above works attempted to incorporate general contextual information in RBAC model, none of them has comprehensively studied the impacts of context on all the components of RBAC model.

We conclude that despite the considerable amount of research on spatio-temporal RBAC models, existing work suffers from significant shortcomings. These include poorly defined authorization semantics, syntax that is both complicated and inadequate, lack of compatibility with RBAC96/ANSI-RBAC standard and a lack of consistency. The GTRBAC model and that of Ray and Toahchoodee – perhaps the two most detailed models in the literature – suffer from all of these problems. We have already noted some of these problems in earlier sections. Comparing the syntactic complexity, Joshi *et al* define 23 predicates in GTRBAC, Ray and Toahchoodee define 16, whereas we supplement RBAC96 with two functions λ and μ . Perhaps the biggest difference between our approach and existing work is to focus on semantics, rather than syntax; we believe the former to be much the harder and less well understood of the two aspects of a spatio-temporal RBAC model.

8. CONCLUSION

In this paper, we constructed a number of spatio-temporal role-based models based on RBAC96 and ERBAC07 using a simple extension of the syntax used for RBAC96 and the ANSI-RBAC standard. We introduced a graph-based formalism to explain the semantics of RBAC96, and used this as a basis for defining the semantics of our spatio-temporal models. We note, in passing, that these semantics might be a useful addition to the ANSI-RBAC standard.

We examined the difficulties that arise when enabling constraints are placed on roles in the presence of role hierarchy. We proposed the use of flat RBAC_{ST}^+ to encode spatio-temporal constraints. When it is necessary to use the role hierarchy, perhaps the most important conclusion of our work is that it is rarely helpful to impose such enabling constraints on roles; instead, these constraints should be applied to users and user-role assignments. We can realize this approach by using RBAC_{ST}^+ and specifying that $\lambda(r) = \mathcal{D}$ for all roles r . We also demonstrated that some pre-computation of enabling conditions on the transitive closure of (part of) RBAC graph can be performed to simplify the evaluation of access requests in the case that enabling conditions are placed on roles.

There are two interesting directions for future work. A first priority is to investigate spatio-temporal separation of duty. We would like to formally classify various spatio-temporal separation of duty constraints, and propose efficient mechanisms for enforcing those constraints.

We also intend to extend the model to any partially ordered set of entity attributes, not just space and time. For example, imagine that there are several security domains within an organization and that each domain is associated with a security clearance. Then some entities/assignments are only enabled when the user belongs to an appropriate domain.

9. ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their helpful comments.

10. REFERENCES

[1] American National Standards Institute. *ANSI INCITS 359-2004 for Role Based Access Control*, 2004.

[2] C. A. Ardagna, M. Cremonini, E. Damiani, S. D. C. di Vimercati, and P. Samarati. Supporting location-based conditions in access control policies. In *Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security*, pages 212–222, 2006.

[3] E. Bertino, P. A. Bonatti, and E. Ferrari. TRBAC: A temporal role-based access control model. *ACM Transactions on Information and System Security*, 4(3):191–233, 2001.

[4] E. Bertino, B. Catania, M. L. Damiani, and P. Perlasca. GEO-RBAC: A spatially aware RBAC. In *Proceedings of the Tenth ACM Symposium on Access Control Models and Technologies*, pages 29–37, 2005.

[5] L. Chen and J. Crampton. Inter-domain role mapping and least privilege. In *Proceedings of the 12th ACM Symposium on Access Control Models and Technologies*, pages 157–162, 2007.

[6] M. J. Covington, W. Long, S. Srinivasan, A. K. Dev, M. Ahamad, and G. D. Abowd. Securing context-aware applications using environment roles. In *Proceedings of the Sixth ACM Symposium on Access Control Models and Technologies*, pages 10–20, 2001.

[7] C. K. Georgiadis, I. Mavridis, G. Pangalos, and R. K. Thomas. Flexible team-based access control using contexts. In *Proceedings of the Sixth ACM Symposium on Access Control Models and Technologies*, pages 21–27, 2001.

[8] F. Hansen and V. Oleshchuk. SRBAC: A spatial role-based access control model for mobile systems. In *Proceedings of the 7th Nordic Workshop on Secure IT Systems*, pages 129–141, 2003.

[9] J. B. D. Joshi, E. Bertino, U. Latif, and A. Ghafoor. A generalized temporal role-based access control model. *IEEE Transactions on Knowledge and Data Engineering*, 17(1):4–23, 2005.

[10] M. Nyanchama and S. Osborn. The role graph model and conflict of interest. *ACM Transactions on Information and System Security*, 2(1):3–33, 1999.

[11] I. Ray and M. Kumar. Towards a location-based mandatory access control model. *Computers & Security*, 25(1):36–44, 2006.

[12] I. Ray and M. Toahchoodee. A spatio-temporal role-based access control model. In *Proceedings of the 21th Annual IFIP WG 11.3 Working Conference on Data and Applications Security*, pages 211–226, 2007.

[13] R. Sandhu. Role activation hierarchies. In *Proceedings of the Third ACM Workshop on Role-Based Access Control*, pages 33–40, 1998.

[14] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, 1996.

[15] M. Strembeck and G. Neumann. An integrated approach to engineer and enforce context constraints in RBAC environments. *ACM Transactions on Information and System Security*, 7(3):392–427, 2004.

[16] R. K. Thomas. Team-based access control (TMAC): A primitive for applying role-based access controls in collaborative environments. In *Proceedings of the Second ACM Workshop on Role-Based Access Control*, pages 13–19, 1997.