SRBAC: A Spatial Role-Based Access Control Model for Mobile Systems

Frode Hansen and Vladimir Oleshchuk

Agder University College,
Department of Information and Communication Technology,
Grooseveien 36, 4876 Grimstad, Norway
{frode.hansen,vladimir.oleshchuk}@hia.no

Abstract. Role-based access control models are receiving increasing attention as a recent generalized approach to access control. In mobile computing environments (that offers location based services), availability of roles and permissions may depend on users location. To cope with the spatial requirements, we extend the existing RBAC model and propose a Spatial Role-based Access Control (SRBAC) model that utilize location information in security policy definitions.

Keywords: Role-Based Access Control, security policy, location information, mobile systems

1 Introduction

Role-Based Access Control (RBAC) models [1, 2] are receiving increasing attention as a recent generalized approach to access control. It differs from traditional identity based access control in that it takes advantage of the concept of role relations. In such models, the user's rights to access computer resources (objects) are determined by the user's membership to roles and by these roles' permissions to perform operations on objects. Thus, a role is a collection of permissions (or operations on a set of objects) determined by the system, based on the users organizational activities and responsibilities, as well as policies for an organization. Therefore, whenever a user has been properly authenticated by the system, this user may activate a subset of roles assigned to the user in order to accomplish his/hers tasks.

The advantages of the concept of roles are several. Firstly, it simplifies authorization administration because a security administrator needs only to revoke and assign the new appropriate role memberships if a user changes its job function. Furthermore, RBAC has shown to be policy neutral [3] and supports security policy objectives as least privilege and static and dynamic separation of duty constraints [2]. Moreover, RBAC offers flexibility with respect to different security policies and in fact [4] shows that RBAC can be configured to enforce mandatory and discretionary access control policies. Recent models [3, 5] extend the RBAC model by specifying temporal constraints on roles that is associated with a user.

Because of the mentioned above, RBAC has been widely investigated. However, even though this great interest for RBAC as way of constraining users access to computer systems and the maturity of models, there are still issues not addressed by the existing RBAC models. One such requirement is related to that the system should be able to base its access decisions depending on the spatial dimension in which the user is situated. The reason for this is that mobile computing devices and wireless networks are increasingly being utilized by organizations. This enables users gaining access to networked computer resources, anywhere and anytime, through their mobile terminal. In organizations where access to resources are limited to a specific location, location-based services require means for obtaining the position of the requesting user in order to mediate the authorization request. Consider, for instance, the case of a doctor that has permission to access a patient's electronic patient record (EPR). However, due to the sensitive information that this EPR contains, the doctor is only authorized to access the EPR in designated areas. Thus, if the doctor request to access a particular patient's EPR from less trustworthy locations such as a hospital cafeteria or reception where there can be a considerable accumulation of people (doctors, nurses, patients, visitors, etc.); the doctor's access request is denied (more information on the application of location based RBAC in healthcare environments can be found in [6, 7]).

In order to cope with the spatial requirements, we propose a Spatial Role-based Access Control (SRBAC) model, an extension of the existing RBAC model proposed in [2], to be able to specify spatial constraints on enabling and disabling of roles. SRBAC support can be used to constrain the set of permissions available to roles that a user may activate at a given location.

Several solutions related to our work have been discussed in the literature. Spatial security policy for mobile agents and mechanisms to provide such policies are discussed in [8]. However, authors do not discuss how it can be used to extend RBAC. In [9, 10], authors extend the RBAC model by introducing the notion of environmental roles in order to control permission sets by (de)activation of roles based on spatial information. The main difference with our work is that in our solution the availability of permission sets depend on spatial information within the same active role. It reduces a number of roles we need to specify within the system and therefore simplify security administration.

The remainder of this paper is organized as follows. In Section 2, we describe the Role-Based Access Control model on which we based our model on together with the formalism used to present location information of a mobile terminal used in the authorization procedure. In Section 3 we present the formal model of SRBAC and finally, Section 4 concludes our work.

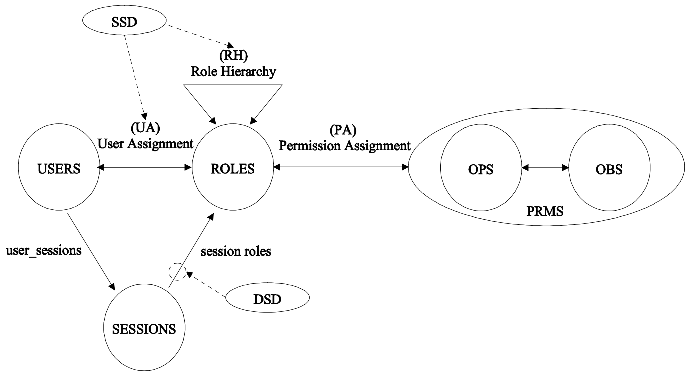


Fig. 1: NIST RBAC Model

2 Preliminaries

This section provides a short description of the standard for Role-Based Access Control (RBAC) proposed by National Institute of Standards and Technology (NIST) [2]. Our model, proposed in Section 3, extend this model by adding the spatial domain component such that authorization decisions can be made also with regards to location. To be able to accomplish this we provide a description of the formalism used to present spatial information about a mobile terminal, which can be used in our Spatial RBAC model.

2.1 RBAC Model

The NIST model, depicted in Figure 1, is defined through four different model components: Core RBAC, Hierarchical RBAC and Static- and Dynamic Separation of Duty Relations. Core RBAC is the base model (minimum requirement) for any RBAC system. Hierarchical RBAC adds the concept of role hierarchies where roles inherit permission from other roles. The two final model components add constraints to the model. The RBAC reference model element sets and relations are further explained below in this section.

Core RBAC Core RBAC encompasses the most essential aspects of Role-Based Access Control and consists of five basic data element sets: $USERS$, $ROLES$, $OBJECTS$ (OBS), $OPERATIONS$ (OPS) and $PERMISSIONS$ (PRMS). Here a user is defined as human beings, machines, networks and autonomous agents. A role is a job function within an organization and permissions are approvals to execute operations on one or more RBAC objects. In addition, the model include sets of $SESSIONS$ where each session is a mapping between a user ($user\_sessions$) and an activated subset of roles ($session\_roles$) that are assigned to the user.

In RBAC, several functions are defined that can be executed on the data element sets. The two relations, the User Assignment (UA) relation and the Permission Assignment (PA) relation, model the assignment of users to roles and the assignment of permissions to roles. Here a user can be assigned to many roles and a role can be assigned to several users. Similarly, a role may be granted several permissions and a permission may be assigned to many roles. Furthermore, the function $user\_session$ associates a session with a single user, and each user is associated with one or more sessions, where a session is a mapping of one user to one or more roles.

Hierarchical RBAC Hierarchies are natural means for structuring roles to reflect the organization's lines of authority and responsibility. In RBAC, role hierarchies define an inheritance relation between roles, denoted by $\succeq$. Such that if $r_i \succeq r_j, r_i, r_j \in ROLES$ then r_i inherits the permissions from role r_j. It is also possible to limit the scope of inheritance by forcing restrictions on the role hierarchy such that inheritance is limited to a single immediate descendant role, denoted $\succ\!\succ$. Thus if $r_i \succ\!\succ r_j$, the role r_i is a single immediate descendant of r_j, if $r_i \succeq r_j$, but no role in the role hierarchy is situated between r_i and r_j.

Constrained RBAC The RBAC model adds the notion of *separation of duty* [2]. A *Separation of Duty* relation is enforced on a set of operations that are mutually exclusive, i.e. no single user may execute all the operations within the set and no single user may be assigned to roles that are conflicting. Constraining the user's actions through the establishment and definition of roles, role hierarchies and role relations may contend this. RBAC accomplish this by enforcing *static separation of duties (SSD)* and *dynamic separation of duties (DSD)*. Through SSD constraints are placed on the assignment of users to roles and especially their ability to form User Assignment associations such that a user may not be authorized for two roles that are mutually exclusive. In addition, it is possible to apply SSD relations in the presence of a role hierarchy. Here, the SSD limitations are inherited such that if a role r_i inherits role r_j and role r_j has an SSD relation with role r_k, this would imply that role r_i also has an SSD relation with role r_k, where $r_i, r_j, r_k \in ROLES$.

Furthermore, introducing DSD relations constrain the permission sets that are available to a user. Contrary to SSD, where constraints are placed on a user's entire permission space, DSD relations restrict the availability of the roles that can be performed simultaneously within a user's session. Therefore DSD allows a user to be authorized to mutually exclusive roles, but these may not be active simultaneously (i.e. DSD relations define constraints on roles that can be activated within a user's session).

2.2 Location Model

For the system to be able to make authorization decisions based on the spatial dimension in which the user is situated, the mediator must be able to obtain the location of the mobile terminal in which the access request was made from. There exist several location-sensing techniques that vary in granularity for both indoor and outdoor position estimation of mobile terminals. The GPS (Global Positioning System) system is a well-known technique for location sensing and can be used to estimate the location of mobile terminals. GPS emits coded radio signals that can be processed in a mobile terminal (with a GPS receiver) to determine its position, time and velocity. The GPS provides high position accuracy and the GPS radio signals can be utilized to compute positions in three-dimensional space. However, since this technique requires line-of-sight of the mobile terminal, it works properly only for outdoor determination of the position of a mobile terminal [11].

For indoor location tracking of mobile terminals one may use location sensing systems such as the Active Badge [12] location system. This system was the first indoor location system developed for use in an office computing environment, and use infrared (IR) technology to keep track of active badges worn by employees. Another solution is to use the 3D-iD system from Pinpoint that use radio frequency (RF) signals and an array of antennas placed at known positions to be able to track a mobile terminal [13]. The Cricket location-support system [14] makes use of both ultra sound and RF signals for the mobile terminals to "*learn their physical location by using listeners that hear and analyze information from beacons spread throughout the building*" [14].

For wireless networks one would have to incorporate more than one of these techniques for location estimation of mobile terminals. The type of location estimation technique used depends on the requirement of accuracy to the mobile terminal' position, which is required by the system in the authorization process. For example, for a user requesting access to a secure service limited to a specific room in a building, may require fine granularity in order to ensure that the user does not try to access the service from the room next door. Therefore, a location system must be able to cope with several location spaces [15]: radio field or infrared cells, access point addresses and geographical coordinates. In addition, due to the diversity in location-spaces, the location information must be represented in a universal and flexible way, such that it can yield for a *ubiquitous computing environment* [15].

In addition to obtaining the location information of the mobile terminal, the system must also be able identify the authenticity of the spatial information obtained. The service that provides the location information used in the authorization process must be able to provide secure and trusted data. This is particularly important for a service which requires precise accuracy of the mobile terminal in order to prevent disclosure of classified information. However, this is beyond the scope of this paper, we assume that the system can identify and verify location of any legitimate user based on a *trusted* underlying network architecture.

In our access control model, in order to ensure this viability, locations are represented by means of symbolic formalism that defines locations as location expressions which describes location areas identifiable by the system.

3 SRBAC Model

As explained earlier, in traditional RBAC, users are assigned to roles and permissions are associated with roles, such that a user may activate permissions dependent on their role assignments. Incorporating traditional RBAC in a mobile environment, one would have to define roles for each location in an organizational domain. Therefore, in organizations where the location domains are many, due to location specific services, roles defined in the system becomes considerable. Moreover, roles defined for these locations may have a lot of the same permissions assigned to them. Thus, in a mobile setting, we can achieve more flexibility defining the security policy when permissions are assigned dynamically to a role limited by the location in which a user is situated. Therefore, a role is dynamic in the sense that it may have different permissions assigned to it for two distinct locations. For example, in a bank a customer is assigned to the role *customer_role*, and has permission to open his/hers safety-deposit box (and of course other permissions related to the nature of such a role). However, the permission to open the safety-deposit box is restricted by the position of the customer, i.e. the system should only grant permission to open the box only when the customer is nearby the safety-deposit box. Thus, the customer may activate his/hers role of a customer when entering the bank, but may not activate this particular permission until entering the strongroom, where the safety-deposit-box is located. If the Figure 2 shows the wireless environment of the bank subdivided into three different zones and $\texttt{Zone}_2$ is the strongroom location.

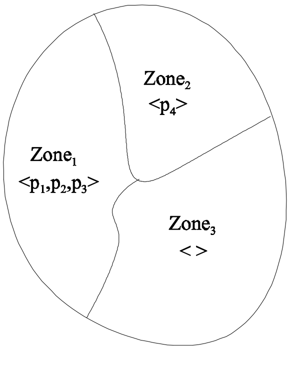


Fig. 2: Logical location domains with available permissions.

For the *customer_role*, permissions associated with it, varies with location such that a user assigned to it has permissions $\texttt{p}_1, \texttt{p}_2, \texttt{p}_3$ in $\texttt{Zone}_1$, only permission $\texttt{p}_4$ in $\texttt{Zone}_2$ and no permissions in $\texttt{Zone}_3$ (denoted $\varnothing$). For example, permission p_4 can indicate that a user assigned to the *customer_role* may open his/hers

Table 1: Location Permission Assignment List (LPAL) where the *customer_role* has different permissions for distinct locations

ROLES	LOC	PERMISSIONS
customer_role	$\mathtt{Zone}_1$	$\mathtt{p}_1$,$\mathtt{p}_2$,$\mathtt{p}_3$
customer_role	$\mathtt{Zone}_2$	$\mathtt{p}_4$
customer_role	$\mathtt{Zone}_3$	$\varnothing$

safety-deposit box only when located in $\mathtt{Zone}_2$. The $\varnothing$ implies that there are no permissions associated with *customer_role* in $Zone_3$, indicating that users associated with the *customer_role* may not access any of the services offered in this particular zone. For a system, the permissions assigned to *customer_role* with regards to locations, can be listed in a *Location Permission Assignment List* (LPAL) as shown in Table 1.

In the remainder of this section we introduce the formal model components of the Spatial Role-Based Access Control (SRBAC) model.

3.1 Core SRBAC

We extend the existing RBAC model [2] to be able to utilize location information in security policy definitions. The SRBAC model consists of the following five basic components: sets $Users$, *Roles*, *Permissions* ($PRMS$), *Sessions* and *Locations* (LOC), representing the set of users, roles, permissions, sessions, and spatial locations respectively. $Users$ are considered to be mobile units that can establish (wireless) communication with system resources to perform some activities. *Roles* are described as a set of permissions to access system resources (objects). *Permissions* are approvals to execute some operation on one or more RBAC objects, and depend on the role and role owner location. *Locations* are represented by means of symbolic expressions called location expressions that describe location domains identifiable by the systems. We assume that wireless network can identify and verify location of any legitimate user based on underlying network architecture (as discussed in Subsection 2.2).

We assume that areas defined in LOC cover the whole responsibility domain Z of SRBAC. The domain Z is divided on the physical layer into subareas, called *primary location cells* denoted as π_i, $i = 1, .., k$, which reflect the ability of the underlying architecture to uniquely map user location into cells. We assume that underlying infrastructure is unable to distinguish different locations inside π_i for any $i = 1, .., k$. However, using primary location cells in SRBAC can be unpractical because primary location cells represent infrastructure of location detection system but we need the structure location domains reflecting organizational infrastructure. Therefore we introduce *logical location domains* that reflect organizational location infrastructure and organizational security policy. For example, within a University we can define logical location domains representing locations such as departments, laboratories and even individual offices. They can be defined as composition of primary cells.

For example, allocation of ICT department can be described by location expression ICT_dom=[π_1, π_3] as area covered by primary location cells π_1 and π_3. Similarly, LIB_dom=[π_2, π_4, π_5] defines library location area. Assuming that CS_dom, EE_dom and IS_dom are logical location domains for departments of Computer Science, Electrical Engineering and Information Science, respectively, we can define domain for School of Computing CSchool_dom as composition of all its departments in the form of location expression, i.e., CSchool_dom=ICT_dom +CS_dom+EE_dom+IS_dom. The example demonstrates the idea of using location expressions to define new domains. Since logical location domains can be seen as sets we define new location domains by using domain operations that are similar to operations used in set theory, i.e., union (denoted as '+'), intersection (denoted as '×'), difference (denoted as '$-$') and complementation (denoted as '$\neg$' or '*outside*'), etc.

Generally, the same position can belong to different logical location domains. In order to simplify definitions and implementations, it is desirable to identify a least set of locations that can be used in location expressions to define all meaningful location domains in SRBAC.

A location l from LOC is called *homogeneous* with respect to role r from *Roles* if r has the same permissions available in any position inside l. Location l from LOC is called *homogeneous* (with respect to *Roles*), if it is homogeneous with respect all r from *Roles*.

Definition 1. *Set of locations $L = \{l_1, l_2, ..., l_k\}$ from LOC are called* normalized *with respect to set of roles R from Roles if it is*

- *a partition of LOC, that is, $LOC = \bigcup_{i=1}^{k} l_i$ and $l_i \cap l_j = \varnothing$ for $i \neq j$, and*
- *any location l_i from LOC is homogeneous with respect to R.*

It is easy to see that any meaningful location expression can be presented as a subset of normalized LOC. From now we assume that LOC is a normalized set of locations (with respect all roles from *Roles*) that is a partition of the entire domain area controlled by SRBAC.

On the sets $Users$, *Roles*, *Permissions* (*PRMS*), *Sessions* and *Locations* (LOC) several functions are defined. The user assignment relation UA, represents the assignment of a user from $Users$ to roles from *Roles*. The permission assignment relation PA, represents the assignment of permissions to roles based on location. We model user assignments to sessions by function *user_sessions* where users can be associated a single session.

Definition 2. *SRBAC model consists of the following components.*

- *USERS, ROLES, PRMS, SESSIONS and LOC, represent the finite set of users, roles, permissions, sessions and locations respectively;*
- *$UA \subseteq USERS \times ROLES$, the relation that associates users with roles;*
- *$assigned\_users(r : ROLES) \rightarrow 2^{USERS}$, the mapping of a role onto a set of users. Formally: $assigned\_users(r) = \{u \in USERS \mid (u, r) \in UA\}$;*

- $PA \subseteq ROLES \times LOC \times PRMS$, *the relation that assigns a permission to a role available in location;*
- $assigned\_permissions(r : ROLES, l : LOC) \rightarrow 2^{PRMS}$, *the mapping of a role r onto a set of permissions based on location. Formally:* $assigned\_permissions(r, l) = \{p \in PRMS | \ (r, l, p) \in PA\}$;
- $user\_sessions(u : USERS) \rightarrow 2^{SESSIONS}$, *assigns a user onto a set of sessions;*
- $session\_roles(s : SESSIONS) \rightarrow 2^{ROLES}$, *the mapping of each session to a set of roles;*
- $avail\_session\_permissions(s : SESSIONS, l : LOC) \rightarrow 2^{PRMS}$, *the permissions available in a session for a location,* $\bigcup\limits_{r \in session\_roles(s)} assigned\_permissions(r, l)$.

3.2 Hierarchical SRBAC

As explained earlier, hierarchies in RBAC define an inheritance relationship between roles, such that a role r_i inherits the permissions from role r_j if all permissions of r_j are also permissions of r_i. Since we present a model where the permissions assigned to the roles varies with location, the permission inheritance relationship among roles in presence of a role hierarchy must also depend on the location. That is, a role r_i would inherit the permissions of role r_j in locations L if all the permissions of r_j in locations L are also permissions of r_i in locations L and if $L \subseteq LOC$.

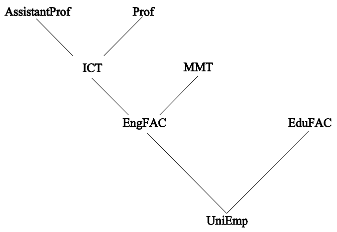


Fig. 3: Role hierarchy example where the role `Prof` would inherit permissions from the roles; `ICT`, `EngFAC` and `UniEmp` in locations specified by L.

This would mean that the role `Prof` in Figure 3 can activate all the permissions inherited from the roles `ICT`, `EngFAC` and `UniEmp` dependent on location together with the permissions assigned to the `Prof` directly.

The location dependent role hierarchy can be formally defined as follows.

Definition 3. *Role Hierarchies in SRBAC*

- $RH \subseteq ROLES \times ROLES \times LOC$ *is a partial order on roles with respect to locations, called dominance relation, written as* $\underset{(L)}{\succeq}$, *where* $r_i \underset{(L)}{\succeq} r_j$, *means* *that role $r_i \in ROLES$ inherits all permissions that role $r_j \in ROLES$ has in locations $L \subseteq LOC$, and all the users of r_i are also users of r_j. If L is omitted then role r_i inherits all the permissions of r_j with respect to locations where r_j is defined;*

- *$auth\_permissions(r : ROLES, l : LOC) \rightarrow 2^{PRMS}$ is the mapping of a role r onto a set of permissions based of location l in presence of a role hierarchy (the permission set assigned directly to the role for that location together with permissions assigned to its junior roles in that location). Formally: $auth\_permissions(r, l) =$*
  $assigned\_permissions(r, l) \cup \left\{ \bigcup_{\forall r' : r \underset{(l)}{\succeq} r'} auth\_permissions(r', l) \right\}$;
- *$auth\_usr(r : ROLES) \rightarrow 2^{USERS}$, the mapping of a role r onto a set of users in presence of a role hierarchy. Formally: $auth\_usr(r) = \{u \in USERS | r' \succeq r, (u, r') \in UA\}$.*
- *Generally: Let L be a set of locations $\{l_1, l_2, ...\}$ normalized with respect to roles $r_i, r_j \in ROLES$ and $l_i \in LOC$. Then $r_i \underset{(L)}{\succeq} r_j$ means $\bigvee_{l \in L} \left(r_i \underset{(l)}{\succeq} r_j \right)$.*

From the above definition follows that if $r_i \underset{(l)}{\succeq} r_j$, then $auth\_permissons(r_j, l) \subseteq auth\_permissons(r_i, l)$ and $auth\_usr(r_i) \subseteq auth\_usr(r_j)$.

3.3 Constrained SRBAC

The proposed RBAC model [2] defines separation of duties to be enforced on a set of roles that may not be executed simultaneously by a user. Our model, extend the concept of separation of duties to allow users to be authorized to mutually exclusive roles if they cannot be executed in the same location. It is similar to SSD and DSD in that intends to limit the permissions available to a user. It differs from SSD and DSD in that the roles are mutually exclusive reliant on the location in which a user is situated. That is, two roles with assigned permissions may be mutually exclusive for a given location, however, for another location a user may be authorized to activate these two roles, since the set of permissions assigned to the roles may be different for distinct locations.

We define in our model both *Spatial Static Separation of Duty* and *Spatial Dynamic Separation of Duty* relations and are further elaborated and defined in the next two subsections.

Spatial Static Separation of Duty Relations. *Spatial Static Separation of Duty relations (SSSD)* enforce constraints on the assignment of users to roles with regards to location. This implies that if a user is assigned to a role in one location, the user cannot be assigned to another role in this location if these to roles are conflicting. Thus, a user may never activate to two roles that share a SSSD relation for a specified location. This is the stronger separation of duty

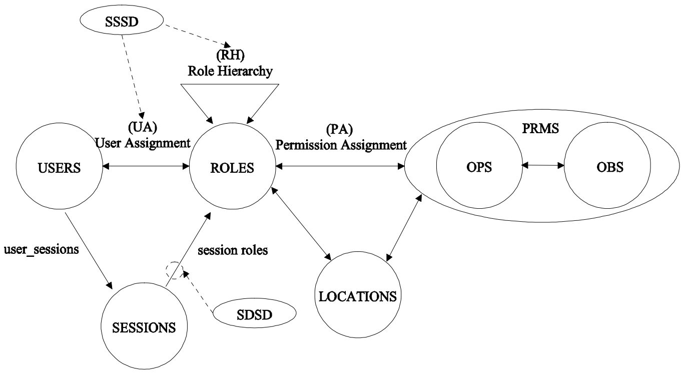
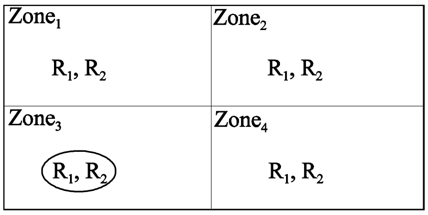


Fig. 4: Spatial Separation of Duty relation

relation, and our model would be similar to the standard RBAC model if the SSSD relation is valid for the entire location space.

Let us illustrate this with an example shown in Figure 5. This example environment contains two roles $\mathtt{R}_1$ and $\mathtt{R}_2$ that a user may activate for all locations except location $\mathtt{Zone}_3$, that is since we assume that $(\langle \mathtt{R}_1, \mathtt{R}_2 \rangle\, ; \mathtt{Zone}_3) \in SSSD$. In classical SSD, enforcing constraints on $\mathtt{R}_1$ and $\mathtt{R}_2$ would result in that no one user can be assigned to both roles for all the locations, that correspond to restriction $(\langle \mathtt{R}_1, \mathtt{R}_2 \rangle\, ; \mathtt{Zone}_1, \mathtt{Zone}_2, \mathtt{Zone}_3, \mathtt{Zone}_4) \in SSSD$ that is same as $(\langle \mathtt{R}_1, \mathtt{R}_2 \rangle) \in SSD$.



Zone_1 R_1, R_2	Zone_2 R_1, R_2
Zone_3 R_1, R_2	Zone_4 R_1, R_2

Fig. 5: Example on SSSD constraint where no one user is allowed to be assigned to both $\mathtt{R}_1$ and $\mathtt{R}_2$ in $\mathtt{Zone}_3$ and an example on Violation of a SDSD relation.

The formal definition of Static *Spatial* Separation of Duty is given below.

Definition 4. *Spatial Static Separation of Duty (SSSD).*

- $SSSD \subseteq \left(2^{ROLES} \times 2^{LOC} \times N\right)$ *is a collection of triples* (rs, ls, n) *where each* rs *is a role set,* ls *is normalized location set, and* n *is a natural number,* $n \geq 2$*, with the property that no user can be assigned to* n *or more roles from the set* rs *in any normalized location* l *from* ls*. Formally:* $\forall\, (rs, ls, n) \in SSSD$, $\forall l \in ls$, $\forall t \subseteq rs : |t| \geq n \Rightarrow \bigcap\limits_{r \in t} authorized\_usr(r, l) = \varnothing$.

Spatial Dynamic Separation of Duty Relations. *Spatial Dynamic Separation of Duty relations (SDSD)* are enforced on the permissions assigned to roles that are activated in a user's session (see Figure 4). SDSD relations allow users to be assigned to two or more roles that are not conflicting when activated in separate sessions for specified locations, however, it would generate policy concerns when activated simultaneously in a user's session for other specified locations. This offers a great advantage compared with classical DSD, due to the fact that one can limit the validity of the constraint to yield in specific locations. A classical DSD constraint enforce restrictions on roles on the entire organization, i.e., in our case, the whole location space, while SDSD, limit the constraint only to be valid dependent on location such that a user may activate conflicting roles within a session for a location, other than the location where the SDSD constraint is specified. Let illustrate with an example.

In Figure 5, we see an example of a wireless environment where a user takes up two roles, $\mathtt{R}_1$ and $\mathtt{R}_2 \in ROLES$. These two roles may are authorized to be activated dependent of location, on various resources offered in this wireless environment. In addition, no one user is allowed to activate both $\mathtt{R}_1$ and $\mathtt{R}_2$ in location $\mathtt{Zone}_3$ in a single session, $(\langle \mathtt{R}_1, \mathtt{R}_2 \rangle ; \mathtt{Zone}_3) \in SDSD$. No SDSD constraint on $\mathtt{R}_1$ and $\mathtt{R}_2$ are specified for locations $\mathtt{Zone}_1$, $\mathtt{Zone}_2$, $\mathtt{Zone}_4$, thus a user may activate these two roles in a single session for the three locations. However, in $\mathtt{Zone}_3$, the user would violate the spatial separation of duty constraint defined in the security policy if the user was to activate $\mathtt{R}_1$ and $\mathtt{R}_2$ in one session (marked by an ellipse in Figure 5). Therefore, the user is not capable of activating both these roles in the same session in location $\mathtt{Zone}_3$. In classical DSD, the constraint on $\mathtt{R}_1$ and $\mathtt{R}_2$ would not only apply to $\mathtt{Zone}_3$, but the entire location space, consisting of $\mathtt{Zone}_1$, $\mathtt{Zone}_2$, $\mathtt{Zone}_3$ and $\mathtt{Zone}_4$.

The formal definition of Dynamic Spatial Separation of Duty is given below.

Definition 5. *Spatial Dynamic Separation of Duty (SDSD).*

- $SDSD \subseteq \left(2^{ROLES} \times 2^{LOC} \times N\right)$ *is a collection of triples* (rs, ls, n) *where each* rs *is a role set,* ls *is normalized location set, and* n *is a natural number,* $n \geq 2$*, with the property that no user may activate* n *or more roles from the set* rs *in any normalized location* l *from* ls*. Formally:* $\forall\, (rs, ls, n) \in SDSD$, $\forall l \in ls$, $\forall s \in SESSIONS$,
  $\forall t \subseteq session\_roles(s) \cap rs : |t| \geq n \Rightarrow \bigcap_{r \in t} authorized\_usr(r, l) = \varnothing$.

4 Conclusions

In this paper we have presented Spatial RBAC (SRBAC), a novel model that extends RBAC to incorporate location information associated with roles in order to permit location-based definition of security policy. In the SRBAC model, permissions are dynamically assigned to the role dependent on location, thus a user assigned to a role may have different permissions reliant on the location. Incorporating spatial information in RBAC as proposed in this paper would enable RBAC to implemented in future mobile computing environments.

References

1. Sandhu, R.S., Coyne, E.J., Feinstein, H.L., Youman, C.E.: Role-based access control models. IEEE Computer **29** (1996) 38–47
2. Ferraiolo, D.F., Sandhu, R., Gavrila, S., Kuhn, D.R., Chandramouli, R.: Proposed NIST standard for role-based access control. ACM Transactions on Information and System Security (TISSEC) **4** (2001) 224–274
3. Bertino, E., Bonatti, P.A., Ferrari, E.: TRBAC: A temporal role-based access control model. ACM Transactions on Information and System Security **4** (2001) 191–223
4. Osborn, S., Sandhu, R., Munawer, Q.: Configuring role-based access control to enforce mandatory and discretionary access control policies. ACM Transactions on Information and System Security (TISSEC) **3** (2000) 85–106
5. Joshi, J.B.D., Bertino, E., Latif, U., Ghafoor, A.: Generalized temporal role based access control model (GTRBAC) (part I)– specification and modeling. Technical report, CERIAS TR 2001-47, Purdue University, USA (2001)
6. Hansen, F., Oleshchuk, V.: Spatial role-based access control model for wireless networks. In: IEEE Vehicular Technology Conference VTC2003. (2003)
7. Hansen, F., Oleshchuk, V.: Application of role-based access control in wireless healthcare information systems. In: Proc. For Scandinavian Conference in Health Informatics. (2003) 30–33
8. Scott, D., Beresford, A., Mycroft, A.: Spatial security policies for mobile agents in a sentient computing environment. In: Lecture Notes in Computer Science, Springer-Verlag Heidelberg (2003) 102–117
9. Covington, M.J., Long, W., Srinivasan, S., Dev, A.K., Ahamad, M., Abowd, G.D.: Securing context-aware applications using environment roles. In: Proceedings of the sixth ACM symposium on Access control models and technologies, ACM Press (2001) 10–20
10. Mantoro, T., Johnson, C.: Location history in a low-cost context awareness environment. In: Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003, Australian Computer Society, Inc. (2003) 153–158
11. Kaplan, E.: Understanding GPS: Principles and Applications. Boston: Artech house Publishers (1996)
12. Want, R., Hopper, A., Falcão, V., Gibbons, J.: The active badge location system. ACM Transactions on Information Systems (TOIS) **10** (1992) 91–102
13. Werb, J., Lanzl, C.: A positioning system for finding things indoors. IEEE Spectrum **35** (1998) 71–78
14. Priyantha, N.B., Chakraborty, A., Balakrishnan, H.: The cricket location-support system. In: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, ACM Press (2000) 32–43
15. Leonhardt, U., Magee, J.: Towards a general location service for mobile environments. In: Proceedings of the 3rd IEEE Workshop on Services in Distributed and Networked Environments. (1996) 43–50