

Inter-domain Role Mapping and Least Privilege

Liang Chen
Information Security Group
Royal Holloway, University of London
l.chen-2@rhul.ac.uk

Jason Crampton
Information Security Group
Royal Holloway, University of London
jason.crampton@rhul.ac.uk

ABSTRACT

The principle of least privilege is a well known design principle to which access control models and systems should adhere. In the context of role-based access control, the principle of least privilege can be implemented through the use of sessions. In this paper, we first define a family of simple role-based models that provide support for multiple hierarchies and temporal constraints. We then investigate a question related to sessions in each of these models: the inter-domain role mapping problem. The question has previously been defined and analyzed in the context of a particular role-based model. We re-define the question and analyze it in the context of a number of different role-based models.

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

Algorithms, Security, Theory

Keywords

IDRM, Least privilege, RBAC

1. INTRODUCTION

Role-based access control (RBAC) has been the subject of considerable research in the last decade [1, 2, 4, 8, 12, 13] and is widely accepted as an alternative to traditional discretionary and mandatory access controls. Perhaps the most distinctive and important feature of the RBAC approach is the role hierarchy, which serves two distinct purposes.

- A role is assumed to inherit the permissions assigned to roles below it in the hierarchy; this is called the (*permission*) *usage* aspect of role hierarchy.

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.

SACMAT'07, June 20-22, 2007, Sophia Antipolis, France.

Copyright 2007 ACM 978-1-59593-745-2/07/0006 ...\$5.00.

- A user assigned to a particular role can also activate any subordinate roles in the hierarchy; this is called the *activation* aspect of role hierarchy.

It has been observed that the single role hierarchy for role activation and permission usage in RBAC approach presents some interoperability problems with separation of duty constraints [12] and makes it more difficult to realize certain fine-grained organizational control goals [8, 9]. In an effort to address these issues, Sandhu introduced an extended RBAC model (ERBAC96) [12], in which two different orderings were defined on the set of roles, one governing role activation and one governing permission inheritance. The resulting hierarchies were respectively called the *activation* and *usage* hierarchies.

Temporal constraints, which limit the times during which a role is enabled, have recently attracted considerable interests [2, 8]. It is natural to limit the hours during which the role of night nurse can be activated, for example. Temporal constraints are a central part of the temporal RBAC (TRBAC) [2] and generalized TRBAC (GTRBAC) [8] models. The GTRBAC model also distinguishes between permission usage and role activation by defining different orderings on the set of roles.

The principle of least privilege is one of the most important principles in the design of protection mechanisms for secure computer systems [11]. Whenever possible, a user should be given no more access to resources than is required to complete the task at hand. Basically, the computer system should be able to determine the minimum set of privileges required for the user to perform the task and guarantee that the user is only granted those privileges and no more.

In loosely-coupled distributed environments, users' identities are usually not known in advance to resource owners. Piromrueen and Joshi [10] propose a requirement-driven interoperation approach that maps requests from users in an external domain to RBAC policies in the target domain. Hence, the underlying problem is to find a set of hierarchically related roles in the target domain that are authorized for the requested set of permissions. In order to observe the principle of least privilege, we want the set of roles to be *minimal* (in some suitably defined sense) and the set of permissions acquired by activating that set of roles to *approximate* the set of requested permissions as closely as possible. Du and Joshi [5] refer to this problem as the *inter-domain role mapping* (IDRM) problem. Their statement of the IDRM problem is to find set of roles of minimal cardinality such that the authorized permissions for that set of roles is precisely the set of requested permissions. It is easy

to show that this version of the problem is not well-defined.

In this paper we study the IDR problem in the context of a variety of different RBAC models. We first define three different RBAC models: ERBAC07, which is essentially identical to ERBAC96; TRBAC07, which is a simple temporal model; and ETRBAC07, which is a combination of the previous two models. These simple, yet expressive, RBAC models have clear, well-defined semantics and provide a number of different contexts for studying the problem at hand. Previous work in this area has concentrated on a very limited version of GTRBAC (in which temporal considerations were completely ignored).

We provide a more accurate formulation of the IDR problem and show that it is closely related to the weighted set cover problem, a generalization of the well-known set cover optimization problem. We adopt an incremental approach, first solving the problem in the context of the well known RBAC96 model, and then extending the techniques to models with multiple hierarchies (ERBAC07 and ETRBAC07) and models with temporal constraints (TRBAC07 and ETRBAC07).

The rest of the paper is organized as follows. In the next section, we recall the basic features of RBAC96. We also formally present ERBAC07, TRBAC07 and ETRBAC07, and briefly compare our models with related work in the literature. In Section 3, we define the IDR problem and develop solutions for IDR problem in the context of RBAC96, ERBAC07, TRBAC07 and ETRBAC07. In Section 4, we study the IDR problem from a different angle, in which safety, rather than least privilege, is paramount. Section 5 concludes the paper with some suggestions for future work.

2. A FAMILY OF RBAC MODELS

2.1 RBAC96

The RBAC96 model defines a set of roles R , a role hierarchy $RH \subseteq R \times R$, a user-role assignment relation $UA \subseteq U \times R$ (where U is a set of users), and a permission-role assignment relation $PA \subseteq P \times R$ (where P is a set of permissions). We write \leq to denote the transitive reflexive closure of the RH relation; (R, \leq) is a partially ordered set (since the directed graph of the role hierarchy relation is assumed to be acyclic). We represent an RBAC96 system (an instance of the RBAC96 model) as a tuple (RH, UA, PA) .

DEFINITION 1. *A user u may activate a role r if there exists $r' \in R$ such that $(u, r') \in UA$ and $r \leq r'$. A session is a set of roles activated by a user. 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 $r \leq r'$.*

We write $Ass(u)$ for the set of roles to which a user u is explicitly assigned by the UA relation; that is, $Ass(u) = \{r \in R : (u, r) \in UA\}$. We write $Act(u) \subseteq R$ for the set of roles a user u may activate and $Auth(u) \subseteq P$ for the set of permissions for which u is authorized.

Given $r \in R$, we write $Ass(r)$ to denote the set $\{p \in P : (p, r) \in PA\}$ and $Auth(r)$ to denote the set of permissions for which r is authorized. Given $S \subseteq R$, we write $Auth(S) \subseteq P$ to denote the set of permissions for which the roles in S are authorized.

Given $S \subseteq R$, we write $\downarrow S$ to denote the set of all elements in R that are less than some element in S . That is, $\downarrow S =$

$\{r \in R : \exists s \in S, r \leq s\}$. We define $\uparrow S$ in an analogous fashion. Hence, for example, in RBAC96, $Act(u) = \downarrow Ass(u)$ and $Auth(r) = Ass(\downarrow r)$.

2.2 ERBAC07

We replace the standard role hierarchy relation, with a new relation $RH \subseteq R \times R \times \{a, u\}$. Let $RH_a = \{(r, r') : (r, r', a) \in RH\}$ and $RH_u = \{(r, r') : (r, r', u) \in RH\}$. We call RH_a the *activation hierarchy* and RH_u the *(permission) usage hierarchy*. 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 . We represent an ERBAC07 system as a tuple (RH_a, RH_u, UA, PA) .

DEFINITION 2. *A user u may activate a role r if there exists $r' \in R$ such that $(u, r') \in UA$ and $r \leq_a r'$. 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 $r \leq_u r'$.*

Given $S \subseteq R$, we define $\downarrow_a S = \{r \in R : \exists s \in S, r \leq_a s\}$ and we define \downarrow_u, \uparrow_a and \uparrow_u in an analogous fashion. Hence, in ERBAC07, for example, $Act(u) = \downarrow_a Ass(u)$.

2.3 TRBAC07

We now introduce temporal constraints, which restrict the times during which a role may be activated. We introduce a relation $R_T \subseteq R \times \mathcal{I}$, where \mathcal{I} represents the set of time intervals. The interpretation of $(r, I) \in R_T$ is that role r is *enabled* for all time points belonging to I . We describe one possible representation of elements of \mathcal{I} in the appendix. We represent a TRBAC07 system as a tuple (R_T, RH, UA, PA) .

DEFINITION 3. *A user u may activate a role r at time t if there exists $r' \in R$ and $(r, I) \in R_T$ such that: $r \leq r'$, $(u, r') \in UA$, and $t \in I$. A user u is authorized for permission p at time t if there exists $r' \in R$ and $(r, I) \in R_T$ such that: u may activate r' , $r \leq r'$, $(p, r) \in PA$, and $t \in I$.*

In other words, users may only activate roles that are enabled, and a request for a permission can only be granted if the permission is assigned to an enabled role. It is possible to define other semantics for permission authorization: the most obvious is that u is authorized for p at time t if u may activate r' , $r \leq r'$, $(p, r) \in PA$ and $(r'', I) \in R_T$, for all r'' such that $r \leq r'' \leq r'$. We choose to adopt the simplest semantics. We do not consider temporal constraints on user-role assignment, permission-role assignment, or on role-role assignment (edges in the role hierarchy).¹

We write $Act(u, t) \subseteq R$ for the set of roles a user u may activate at time t and $Auth(u, t) \subseteq P$ for the set of permissions for which u is authorized at t .

2.4 ETRBAC07

This model combines the features of ERBAC07 and TRBAC07. In other words, we have the extended hierarchy relation $RH \subseteq R \times R \times \{a, u\}$ and the temporal role relation $R_T \subseteq R \times \mathcal{I}$. The user-role and permission-role assignment relations remain unchanged. We represent a ETRBAC07 system as a tuple $(R_T, RH_a, RH_u, UA, PA)$.

¹It is easy to define the syntax for such constraints, but it is quite difficult to specify appropriate semantics. The GTRBAC model, for example, defines such constraints, but they have never been used in the analysis of the model.

2.5 Related models

Sandhu defined the ERBAC96 model [12]. ERBAC96 is identical to ERBAC07, with the exception that the ERBAC96 model has the following constraint: $r \leq_a r'$ implies that $r \leq_a r'$.

Like ERBAC96, the GTRBAC model [8] defines a usage and activation hierarchy; unlike ERBAC96, GTRBAC imposes no constraints on these hierarchies. In addition, the GTRBAC model defines a permission-activation hierarchy, but this is redundant and can be defined in terms of the two other hierarchies. Figure 1(a) shows a “hybrid” hierarchy used previously to illustrate the GTRBAC model [5]: a solid edge denotes an element of the usage hierarchy; a dashed edge indicates an element of the activation hierarchy; and a double-headed arrow indicates an element of the permission-activation hierarchy. It is obvious that the hybrid hierarchy can be simply expressed by the activation and usage hierarchies shown in Figure 1(b) and 1(c).

GTRBAC introduced temporal constraints to RBAC. The syntax for the model is rather complicated and the semantics defining the interaction between temporal constraints on the UA , PA , roles, and role hierarchies are not clearly defined. Papers that have appeared subsequently on GTRBAC, have ignored temporal constraints and focused on issues related to the multiple hierarchies [5, 7]. The analysis of the somewhat simplified temporal model we present in this paper represents the first real attempt to consider the effect of temporal constraints on RBAC.

The purpose of defining a family of increasingly complex models is to understand the IDR problem in a very well understood context, such as RBAC96, and then apply the insights obtained to more complex models.

3. INTER-DOMAIN ROLE MAPPING

Du and Joshi define the inter-domain role mapping (IDRM) problem as follows [5, §4.1].

DEFINITION 4. *Given a set of requested permissions $Q \subseteq P$, find the minimal set of roles $R' \subseteq R$ such that $Auth(R') = Q$.*

The interpretation of “minimal” is that the cardinality of R' (henceforth denoted $|R'|$) should be as small as possible. The first thing to note is that the IDR problem is not well defined, for at least two different reasons.

- There may not be a unique minimal set of roles; that is, there may be several sets R_1, \dots, R_k such that $Auth(R_i) = Q$, $i = 1, \dots, k$. Consider the RBAC96 configuration in Figure 2(a). Let us assume that $Q = \{p_2, p_3, p_4, p_5\}$: then $\{r_4, r_7\}$ and $\{r_5, r_7\}$ are both minimal.
- Far more important, however, is the fact that there may not exist $R_Q \subseteq R$ such that $Auth(R_Q) = Q$. Consider the RBAC96 configuration in Figure 2(a) and suppose $Q = \{p_2, p_3, p_4\}$: IDR does not have a solution since any subset of roles R' that is authorized for Q is also authorized for additional permissions outside Q .

This may seem like a trivial objection, but it is crucial when we consider the principle of least privilege: we must seek to minimize the additional permissions that are granted.

We now formulate the IDR problem more accurately. There are two aspects to the problem: we must ensure that all requested permissions are available and we must ensure that the principle of least privilege is observed.

DEFINITION 5. *Given an RBAC system, a session $S \subseteq R$ is said to be minimal if for all $S' \subset S$, $Auth(S') \subset Auth(S)$.*

DEFINITION 6. *Given a set of permissions $Q \subseteq P$, we say $S \subseteq R$ is Q -minimal if $Q \subseteq Auth(S)$, and for all $S' \subset S$, $Auth(S') \subset Q$.*

In other words, S is Q -minimal if all permissions in Q are authorized in session S and for any smaller session there is at least one permission in Q that would not be authorized in that session. Clearly, if S is Q -minimal, then S is minimal. Of course, there may be several such Q -minimal sessions.

REMARK 7. *A session S is Q -minimal only if S is an antichain. However, not all antichains are Q -minimal. Let us take the example of RBAC96 configuration in Figure 2(a), in which $Q = \{p_2, p_3, p_4, p_5\}$, then $R' = \{r_4, r_5, r_7\}$ is an antichain such that $Auth(R') \supset Q$; R' , however, is not Q -minimal as $Auth(\{r_4, r_7\}) = Auth(\{r_5, r_7\}) \supset Q$.*

DEFINITION 8. *Given a set of permissions $Q \subseteq P$, we say S is Q -optimal if S is Q -minimal and for all Q -minimal sets $S' \subseteq R$, $|Auth(S)| \leq |Auth(S')|$.*

In other words, S is Q -optimal if it minimizes the number of authorized permissions that are not in Q . Note that there may still be more than one Q -optimal session. Choosing between such sessions is somewhat arbitrary and probably has to be implementation-dependent. Given several Q -optimal sessions, perhaps the best choice would be one with minimal cardinality, although even this criterion does not necessarily identify a unique session.

Note that if $Q = P$, the problem reduces to the IDR problem examined by Du and Joshi [5]. They noted that there exists a very simple reduction from the IDR problem to the *set covering optimization problem* [3].

DEFINITION 9 (SET COVERING OPTIMIZATION PROBLEM). *Given a universe U , a collection \mathcal{C} of subsets of U whose union is U , find a subset $\mathcal{D} \subseteq \mathcal{C}$ such that*

$$U = \bigcup_{D \in \mathcal{D}} D \quad \text{and} \quad |\mathcal{D}| \quad \text{is minimized.}$$

The set covering optimization problem is NP-hard, although there exists a “greedy” algorithm that provides good approximate solutions [6]. Suppose we have chosen $i - 1$ sets from \mathcal{C} , and let $A_{i-1} \subseteq U$ denote set of elements that remain uncovered. At the i th iteration, the greedy algorithm selects a subset C_i from those remaining in \mathcal{C} such that

$$\frac{1}{|C_i \cap A_{i-1}|}$$

is minimized. (Equivalently, the number of uncovered elements contained in C_i is maximized.)

We now consider the problem of finding a Q -optimal session given $Q \subseteq P$. For consistency with Du and Joshi’s work, we continue to refer to this as the IDR problem.

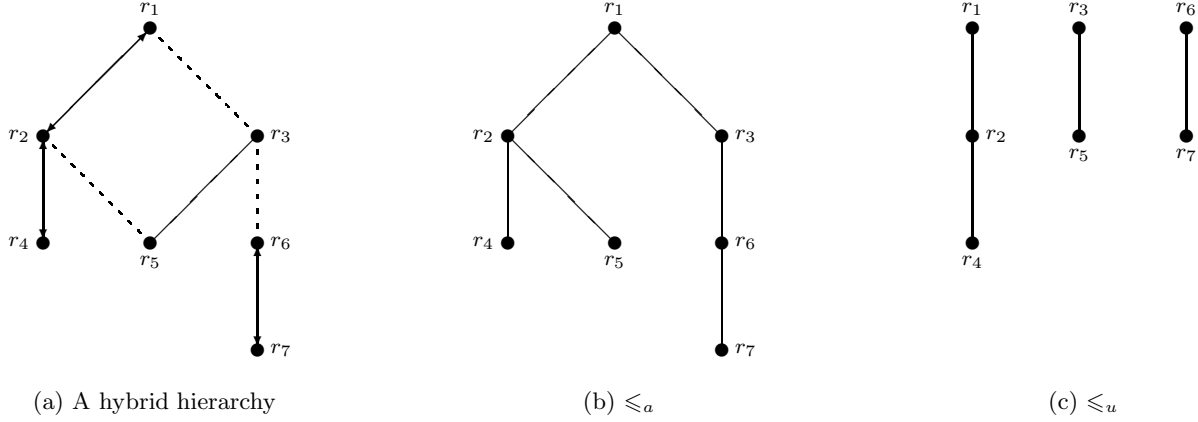


Figure 1: Role hierarchies

3.1 An approximate algorithm for computing IDRMs in RBAC96

Suppose we are given $Q \subseteq P$, PA and RH . For $r \in R$, we define $R_Q = \{r \in R : \text{Auth}(r) \cap Q \neq \emptyset\}$. We want to find a subset of $R' \subseteq R_Q$ such that

1. $\text{Auth}(R') \supseteq Q$,
2. $|\text{Auth}(R')|$ is minimized,
3. R' is a minimal session.

The first condition requires that all permissions in the requested set Q are available to the set R' ; the second condition requires that the number of permissions that are not in Q is minimized, thereby enforcing least privilege.

Take the example of RBAC96 configuration in Figure 2(a) and $Q = \{p_2, p_3, p_4, p_5\}$, we can have the minimal role set $R' = \{r_1\}$, such that $\text{Auth}(r_1) \supseteq Q$. However, we prefer to have the role set $R' = \{r_4, r_7\}$ or $\{r_5, r_7\}$, such that $\text{Auth}(r_4, r_7) = \text{Auth}(r_5, r_7) = Q$.

The mapping between the IDRMs and set covering relies on the assumption that there exists a set of roles R' such that $\text{Auth}(R') = Q$, which in turn assumes that $\text{Auth}(r) \subseteq Q$ for all $r \in R'$. However, the problem that we study is more complex than the IDRMs problem, because we do not assume that $\text{Auth}(r) \subseteq Q$. This means we cannot employ the greedy algorithm used to find a good approximate solution to the set covering optimization problem and the IDRMs problem. Instead, we propose that we map our problem to the *weighted set cover problem* [3].

DEFINITION 10 (WEIGHTED SET COVER PROBLEM).

Given a universe U , a collection \mathcal{C} of subsets of U whose union is U , and a weight function $w : \mathcal{C} \rightarrow \mathbb{R}^+$, find a subset $D \subseteq \mathcal{C}$ such that

$$U = \bigcup_{D \in \mathcal{D}} D \quad \text{and} \quad \sum_{D \in \mathcal{D}} w(D) \quad \text{is minimized.}$$

We define the weight function $w : R_Q \rightarrow \mathbb{R}^+$, where

$$w(r) = |\text{Auth}(r)| \cdot |\text{Auth}(r) \setminus Q| + \frac{1}{|Q|}.$$

Note that $w(r) = \frac{1}{|Q|}$ when $\text{Auth}(r) \subseteq Q$. That is, the weight $w(r)$ is small when $\text{Auth}(r)$ does not contain any elements outside Q , and large otherwise.

We iteratively add elements of R_Q until we have covered all the elements in Q . Let $A_{i-1} \subseteq Q$ denote the set of elements that remain uncovered. We modify the greedy algorithm so that at the i th iteration we choose r such that $A_{i-1} \cap \text{Auth}(r) \neq \emptyset$ and

$$\gamma(r) = \frac{w(r)}{|A_{i-1} \cap \text{Auth}(r)|}$$

is minimized. Clearly, this algorithm runs in time polynomial in $|R_Q|$.

In the special case that $Q = P$, we have that $w(r) = \frac{1}{|Q|}$ for all $r \in R$, which corresponds to the standard set covering optimization problem (since the weight is constant).

EXAMPLE 11. Consider the RBAC96 configuration illustrated in Figure 2(a), where permission-role assignments are indicated by broken lines, and assume that $Q = \{p_2, p_3, p_4\}$. Then

$$\begin{aligned} R_Q &= \{r_1, r_2, r_3, r_4, r_5, r_6, r_7\}, \\ w(r_1) &= \frac{85}{3}, \\ w(r_2) &= w(r_7) = \frac{10}{3}, \\ w(r_3) &= \frac{46}{3}, \\ w(r_4) &= w(r_5) = \frac{1}{3}, \\ w(r_6) &= \frac{25}{3}. \end{aligned}$$

The first iteration of the greedy algorithm selects r_5 since $\gamma(r_5) = \frac{1}{6}$ is minimal. The next iteration of the greedy algorithm selects r_7 , at which point the algorithm terminates. The solution $\{r_5, r_7\}$ is optimal for the IDRMs problem. In fact, the session $\{r_4, r_7\}$ is an optimal solution for the IDRMs problem, and it is also a corresponding optimal solution in the weighted set cover problem. However, the greedy algorithm only generates one optimal solution $\{r_5, r_7\}$.

Nor does the greedy algorithm always produce an optimal solution to the IDRMs problem. Consider the simple RBAC96 configuration shown in Figure 2(b), and assume $Q = \{p_1, p_2, p_3\}$. Then $w(r_1) = \frac{1}{3}$, $w(r_2) = w(r_3) = \frac{7}{3}$, $w(r_4) = \frac{10}{3}$. The first iteration of the greedy

algorithm selects r_1 since $\gamma(r_1) = \frac{1}{3}$ is minimal. The second iteration of the algorithm arbitrarily selects r_5 . Note that r_2 or r_3 could also be selected, as $\gamma(r_2) = \gamma(r_3) = \gamma(r_5) = \frac{7}{3}$. The next iteration of the algorithm selects r_2 , at which point the algorithm terminates. In this case, the greedy algorithm produces the solution $\{r_1, r_2, r_5\}$, whereas $Q \subset \text{Auth}(\{r_1, r_2, r_3\}) \subset \text{Auth}(\{r_1, r_2, r_5\})$. Hence $\{r_1, r_2, r_3\}$ is an optimal solution for the IDR problem.

In summary, the greedy algorithm, although efficient, may only produce a unique optimal solution to the IDR problem, although multiple optimal solutions exist. In addition, the greedy algorithm may not return an optimal solution to the IDR problem. The reason is that the weighed set cover problem assumes there exists an exact cover of Q , and the IDR problem does not. There may be no good trade-off between safety and efficiency for the general IDR problem, except in special cases. We return to this issue in Section 4.

3.2 IDR in ERBAC07

Suppose we are given $Q \subseteq P$, PA , RH_a and RH_u . Recall that $\text{Auth}(r) = \text{Ass}(\downarrow_u r)$ in ERBAC07. Similarly, for $r \in R$, we define $R_Q = \{r \in R : \text{Auth}(r) \cap Q \neq \emptyset\}$. We want to find a subset of $R' \subseteq R_Q$ such that

1. $\text{Auth}(R') \supseteq Q$,
2. $|\text{Auth}(R')|$ is minimized,
3. R' is a minimal session.

Obviously, we can directly employ the greedy algorithm suggested in Section 3.1 to RH_u in order to find an approximate solution to IDR problem in ERBAC07.

3.3 IDR in TRBAC07

Given a TRBAC07 system (R_T, RH, UA, PA) , we have $\text{Auth}(r, t) = \text{Ass}(\downarrow(r, t))$. Suppose the set of permissions Q is requested at a particular point of time t . For $r \in R$, we define $R_Q = \{r \in R : \text{Auth}(r, t) \cap Q \neq \emptyset\}$. We want to find a subset of $R' \subseteq R_Q$ such that

1. $\text{Auth}(R', t) \supseteq Q$,
2. $|\text{Auth}(R', t)|$ is minimized,
3. R' is a minimal session at t .

We can employ the greedy algorithm suggested in Section 3.1 to RH for producing the approximate solution at t to IDR problem in TRBAC07.

3.4 IDR in ETRBAC07

Given an ETRBAC07 system $(R_T, RH_a, RH_u, UA, PA)$, we have $\text{Auth}(r, t) = \text{Ass}(\downarrow_u(r, t))$. Suppose the set of permissions Q is requested at a particular point of time t . For $r \in R$, $R_Q = \{r \in R : \text{Auth}(r, t) \cap Q \neq \emptyset\}$. We want to find a subset of $R' \subseteq R_Q$ such that

1. $\text{Auth}(R', t) \supseteq Q$,
2. $|\text{Auth}(R', t)|$ is minimized,
3. R' is a minimal session at t .

The greedy algorithm can be used to compute approximate solution at t in RH_u for solving the IDR problem in ETRBAC07.

4. A SAFE APPROACH TO IDR

The approach to the IDR problem in the previous section emphasizes *availability*: a solution to the problem results in all the permissions in Q being authorized. One might argue from a conservative viewpoint that it would be *safer* to compute a set of roles R' such that $\text{Auth}(R') \subseteq Q$ and $|\text{Auth}(R')|$ is maximized. In other words, we do not wish to allow any permissions that are not in Q , but want to authorize as many permissions in Q as possible.

DEFINITION 12. *Given a set of permissions $Q \subseteq P$, we say S is Q -safe if S is minimal, $Q \supseteq \text{Auth}(S)$ and for all $S' \supset S$, $\text{Auth}(S') \supset Q$. We say S is Q -optimally-safe if for all Q -safe sets S' , $|\text{Auth}(S)| \geq |\text{Auth}(S')|$.*

Given $Q \subseteq P$, PA and RH , we define $R_Q = \{r \in R : \text{Auth}(r) \subseteq Q\}$. Clearly, $\text{Auth}(R_Q) \subseteq Q$ and for any $R' \subseteq R_Q$, $\text{Auth}(R') \subseteq \text{Auth}(R_Q)$. Hence, $|\text{Auth}(R_Q)|$ is maximized. However, R_Q may not be minimal. Hence, we have the following problem: given a collection of permissions $Q' = \text{Auth}(R_Q)$, a collection of subsets $\{\text{Auth}(r) : r \in R_Q\}$, find $R'_Q \subseteq R_Q$ such that

$$\bigcup_{r \in R'_Q} \text{Auth}(r) = \text{Auth}(R_Q) \quad \text{and} \quad |R'_Q| \text{ is minimized.}$$

In other words, we have an instance of the set covering optimization problem and we can use a slightly modified version of the greedy algorithm for the set covering problem to solve this problem.

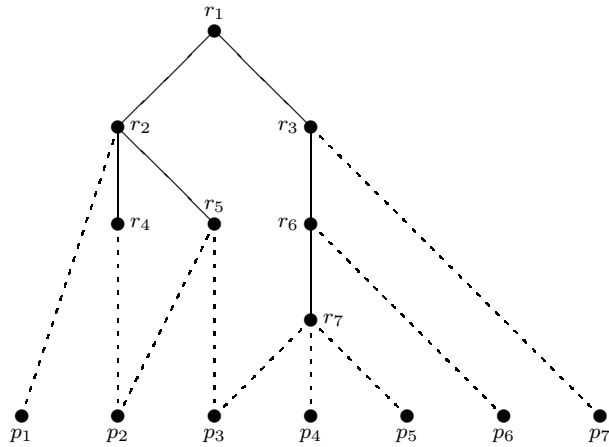
In summary, if we adopt a safe approach to the IDR problem, in which no permissions outside Q are granted, we can obtain a good approximate solution in polynomial time. However, we cannot guarantee that the requester will be authorized for all the permissions she requested.

5. CONCLUSION

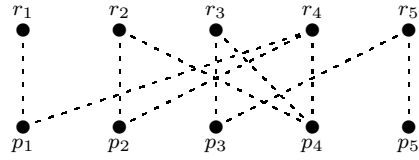
In this paper, we construct a series of role-based models based on RBAC96, and influenced by existing models such as ERBAC96 and GTRBAC. We use these models to investigate least privilege and the IDR problem in the presence of multiple role hierarchies and temporal constraints. The syntax we have chosen for ERBAC07, TRBAC07 and ETRBAC07 is consistent with RBAC96, and the semantics for these models are easy to understand.

We have conducted an analysis of the IDR problem from two different perspectives. In terms of the principle of least principle and availability, we want to find a minimal set of roles that simultaneously authorize the set of requested permissions and minimize the number of additional permissions that are granted. Alternatively, from the point of view of safety, we want to find a minimal set of roles that grant the maximal set of permissions strictly contained in the requested permissions. In this case, the principle of fail-safe defaults is adopted, because a user might not be granted permissions that she requested. For both approaches, the greedy algorithm can be used to provide a good approximate solution to the IDR problem in the context of the different models we developed.

One question that has yet to be resolved to our satisfaction is to find an alternative heuristic for computing a solution to the IDR problem. The greedy algorithm we use in Section 3 may not compute optimal solutions. From



(a) RBAC96 configuration



(b) Simple RBAC96 configuration

Figure 2: RBAC configurations

a security perspective, the greedy algorithm may fail to find a safe solution. It will be interesting to see if there exist algorithms that run in polynomial time and always return a safe solution.

6. ACKNOWLEDGMENTS

We would like to thank Ninghui Li for his valuable comments and suggestions.

7. REFERENCES

- [1] American National Standards Institute. *ANSI INCITS 359-2004 for Role Based Access Control*, 2004.
- [2] 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.
- [3] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, second edition, 2001.
- [4] J. Crampton. On permissions, inheritance and role hierarchies. In *Proceedings of the 10th ACM Conference on Computer and Communications Security*, pages 85–92, 2003.
- [5] S. Du and J. B. D. Joshi. Supporting authorization query and inter-domain role mapping in presence of hybrid role hierarchy. In *Proceedings of the eleventh ACM symposium on Access control models and technologies*, pages 228–236, 2006.
- [6] D. S. Johnson. Approximation algorithms for combinatorial problems. In *Proceedings of the fifth annual ACM symposium on Theory of computing*, pages 38–49, 1973.
- [7] J. B. D. Joshi, E. Bertino, and A. Ghafoor. Hybrid role hierarchy for generalized temporal role based access control model. In *Proceedings of the 26th International Computer Software and Applications Conference on Prolonging Software Life: Development and Redevelopment*, pages 951–956, 2002.
- [8] 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.
- [9] J. D. Moffett and E. C. Lupu. The uses of role hierarchies in access control. In *Proceedings of the fourth ACM workshop on Role-based access control*, pages 153–160, 1999.
- [10] S. Piromruen and J. B. D. Joshi. An RBAC framework for time constrained secure interoperation in multi-domain environments. In *Proceedings of the 10th IEEE International Workshop on Object-Oriented Real-Time Dependable Systems*, pages 36–48, 2005.
- [11] J. Saltzer and M. Schroeder. The protection of information in computer systems. *Proceeding of the IEEE*, 63(9):1278–1308, 1975.
- [12] R. Sandhu. Role activation hierarchies. In *Proceedings of the third ACM workshop on Role-based access control*, pages 33–40, 1998.
- [13] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, 1996.

APPENDIX

A. REPRESENTING TIME INTERVALS

We assume the existence of a clock, whose ticks are indexed by the natural numbers \mathbb{N} . A *time interval* $i = [t_1, t_2]$, where $t_1, t_2 \in \mathbb{N}$ and $t_1 < t_2$, is the set $\{t \in \mathbb{N} : t_1 \leq t \leq t_2\}$. A *complex time interval* I has the form $\bigcup_{j \in J} i_j$, where $J \subseteq \mathbb{N}$ and i_j is a time interval (for all $j \in J$). In other words, a complex time interval is the union of 0 or more time intervals. Let \mathbb{I} denote the set of complex time intervals.

We write $t \in I$ to denote that a particular point in time t belongs to at least one of the intervals contained in I . We write $t \in I_1 \cap \dots \cap I_n$ to denote that $t \in I_1, \dots, t \in I_n$, where $I_j \in \mathbb{I}$. We write $i \in I$ to denote that a particular time interval i belongs to at least one of the intervals contained in I . We write $i \in I_1 \cap \dots \cap I_n$ to denote that $i \in I_1, \dots, i \in I_n$, where $I_j \in \mathbb{I}$. We write $[t_1, t_2] + t$ to denote the interval $[t_1 + t, t_2 + t]$, $I + t$, where $I = \bigcup_{j \in J} i_j$, to denote the complex interval $\bigcup_{j \in J} (i_j + t)$, and $(I_1 \cap \dots \cap I_n) + t$ to denote the complex interval $I_1 + t \cap \dots \cap I_n + t$.