Mandatory Access Control and Role-Based Access Control Revisited

Sylvia Osborn
Department of Computer Science
The University of Western Ontario
London, Ontario, Canada N6A-5B7
email: sylvia@csd.uwo.ca

Abstract

In this paper we reexamine the interaction between role-based access control and mandatory access control. We examine the question: from the perspective of a given role graph in which the objects have been assigned security classifications, can its roles be assigned to subjects without violating mandatory access control rules? A detailed study of the structure of individual roles and edges in a role graph is undertaken. We show that the combination of the structure imposed by the role graphs and the MAC rules means that the possible structure of a role graph in which roles are assignable to subjects without violating MAC rules is greatly restricted.

1 Introduction

Role based access control has been touted as being capable of representing many kinds of security policies and security models. One such security model is Mandatory Access Control (MAC), commonly represented by the Bell-La Padula model [1].

The role graph model of Nyanchama [3, 4] is one example of a role based access control model [8, 9]. In a previous paper [5], we discussed the interaction between roles and mandatory access control. This discussion was of a general nature, looking at the interaction from the role graph perspective, i.e. given certain structures in a role graph, will they guarantee MAC.

In [7], Sandhu goes in the other direction, examining different kinds of lattice-based access control models and translating each of them into a role hierarchy. One observation of the resulting role hierarchies is that they are not very interesting or rich in structure.

The purpose of this paper is to look again at the details of the role graphs of the Nyanchama model. In particular, we look at the details of a single role or node, and a given edge, and determine under what conditions such structures do or do not violate the constraints imposed by mandatory access control. In the end, we give some examples of role graph structures that satisfy MAC constraints. We also see that the possible structures are not very interesting, but the detailed study helps us to understand why this must be so.

We begin by briefly reviewing the role graph model, then go on to a brief description of MAC. The bulk of the paper examines in great detail the possible structure of individual roles and edges when MAC is adhered to.

RBAC 97 Fairfax Va USA
Copyright 1997 ACM 0-89791-985-8/97/11..$3.50

2 The Role Graph Model

The role graph model is based on the notion of users, which for this paper we can call subjects, privileges and roles. In its most general definition, a privilege has been viewed as the combination of an object and a set of operations on the object. More precisely, a *privilege* is a pair (x, m) where x refers to an object, and m is a non-empty set of access modes for object x.

A *role* is defined as a named set of privileges. It is represented by a pair $(rname, rpset)$, where *rname* is the name of the role, and *rpset* represents the set of privileges of the role. Given a role R, we will use $R.rname$ and $R.rpset$ to refer to the role's name and privilege set, respectively.

The roles form the nodes of the role graph. An edge $R_1 \rightarrow R_2$ in the role graph represents the fact that R_1 *is junior to* R_2. Role R_1 *is junior to* R_2 iff $R_1.rpset \subseteq R_2.rpset$.

There are three relationships which determine access control:

- the assignment of privileges to roles. This corresponds to the PA relation in Sandhu's model [7].
- the assignment of users to roles. This corresponds to the UA relation in Sandhu's model [7].
- the edges in the role graph. Edges exist in the role graph because of the way the role graph is created using the role graph algorithms. In the algorithms described in [4], new roles can be defined by giving the privileges for the new role, and the edges to and from its immediate neighbours in the role graph. In the current version of the prototype, two additional operations on the role graph can cause new edges in the graph: a single edge can be added, or a new role can be defined by simply giving its total privileges. In the latter case, the algorithm works out what edges should be in the graph.

  The edges in the role graph correspond to the RH relation in Sandhu's model [7].

Our model currently has no formal model of constraints and no notion of sessions.

Our model also includes a MaxRole and a MinRole. MaxRole represents the union of all the privileges of the roles in the role graph. MaxRole does not need to have any users authorized to it. It is in the role graph to have a place to summarize all of the privileges in the system, and to ensure that the common senior relationship [4] is always defined. MinRole represents the minimum set of privileges available to all roles. MinRole.*rpset* can be empty. Role graphs have the following *Role Graph Properties*:

1. There is a single MaxRole.
2. There is a single MinRole.
3. The Role Graph is acyclic.
4. There is a path from MinRole to every r_i .
5. There is a path from every r_i to MaxRole.
6. For any two roles r_i and r_j, if $r_i.rpset \subseteq r_j.rpset$, then there must be a path from r_i to r_j.

In what we will talk about in this paper, it is not necessary to insist that MaxRole and MinRole be present in the role graph. It is only necessary that properties 3 and 6 above hold. Property 3, acyclicity, is necessary, we feel, so that roles offer differentiated access to the objects in the system. Since we have a well developed and efficient set of algorithms for manipulating role graphs, roles can be added to and deleted from a system very quickly. Therefore, we do not feel that this property unduly restricts the design of a security system.

It is important to note that role graphs can be arbitrarily complex; in particular, they can have arbitrarily many roles in a path from MinRole to MaxRole. The assignment of privileges to roles can also be very complex, and can overlap in arbitrary ways.

3 Mandatory Access Control

We describe here the Mandatory Access Control model commonly known as the Bell-La Padula model. There is a set of *classifications*, (e.g. top secret, secret, confidential, classified,) which is totally ordered. There is, in addition, a set of *categories* that is unordered. The combination of a classification and a subset of the categories is called a *security level.* Security levels are partially ordered and form a lattice [6].

Every subject and object in the system must be labelled by a security level. We will distinguish between *trusted* and *untrusted* subjects as defined in [2]. Trusted subjects can be relied on not to compromise security; all other subjects are untrusted. In the case of subjects, the label is called the *clearance*, and for an object, the label is called the *security classification.* We will denote the security label by $\lambda(s)$ or $\lambda(o)$.

To ensure secrecy, the following two mandatory rules must be followed:

- Simple Security Property: Subject s can read object o only if $\lambda(s) \geq \lambda(o)$.

- $\star$-property: Untrusted subject s can write object o only if $\lambda(s) \leq \lambda(o)$.

The Simple Security Property is sometimes referred to as the "no read up" rule, and the $\star$-property is known as the "no write down" rule.

In the discussion in the next section, we will first assume that the security levels are totally ordered, i.e. that there exist n security levels $\lambda_1, \ldots, \lambda_n$ such that $\lambda_1 > \lambda_2 > \ldots > \lambda_n$. Subsequently, we will reexamine the results with a security lattice which is not totally ordered.

4 Role Graph Constraints to Satisfy MAC

In this section we consider the following problem: given that a role graph has been designed, perform a static analysis on the role graph, where each subject and each object has assigned to it a security level, and determine whether or not any violations of the above two MAC rules exist.

Since the unit of assignment of access control in the role graph model is the role, we need to examine the read and write behaviour of a role in order to analyze a given role graph for adherence to MAC. In a complex system, a privilege might be something like "Hire a new employee". We must assume that for each privilege in each role, an analysis can be performed which transforms all of these privileges into a list of (object, operation) pairs, where the operation is either read or write. Let us call the resulting privilege set the *modified privilege set.*

In the Role Graph Model, when a subject is assigned to a role, that subject can perform **all** the privileges in the role, which include all the privileges in any role junior to this role in the role graph (i.e. from any role for which there is a path in the role graph to the role in question). When analyzing this situation for adherence to the two MAC properties, we must ensure that should the subject in fact execute all the privileges available in a role, no violations of secrecy can occur. Because a role is the minimum granularity at which subjects are assigned any rights, we only consider the case where the subject is at a single clearance, which is the clearance we use to determine whether or not this assignment violates any MAC properties.

4.1 Assignment of Single Roles

Let us now consider a single role R. All objects o for which (o, r) (r stands for read and w for write) is in the modified privilege set for R are said to be in the *r-scope* of R. Similarly, all objects o for which (o, w) is in the modified privilege set for R are said to be in the *w-scope* of R.

Theoretically, it is possible for a role to contain objects at different security levels. Therefore, we define the *r-level* of a role as the **maximum** security level of any object in the role's r-scope. The *w-level* of a role is the **minimum** security level of any object in the role's w-scope.

Consider first a "read-only" role, i.e. a role which has an empty w-scope. If all the objects in r-scope are a the same security level, then this role can be assigned to subjects whose clearance adheres to the Simple Security Property, i.e. to subjects whose clearance dominates this single security level. If the objects in r-scope have different security levels, then any subject whose clearance dominates the maximum security level of the objects accessible in the role can be assigned to the role. If the clearance of the subject is less than some object in the role, then there will be "read up".

> **Constraint 1:** Any subject s assigned to a role R must have $\lambda(s) \geq r\text{-level}(R)$.

Now consider roles with non-empty r-scope and nonempty w-scope. Some possibilities are shown in Figure 1. If all the objects in a role have their r-scope and w-scope within one level, (Figure 1a, role R3) then subjects assigned to this role must be at this level of clearance to adhere to the MAC properties. This is shown in the figure by the $S \rightarrow$ in the legal clearance level for subjects. If the w-scope and r-scopes contain objects at more than one security level, say as in Figure 1a, role R1, where the overlap is all within one security level, then a subject whose clearance is exactly this overlapping level can be assigned to this role: there will be no read up and no write down, because all of the r-scope is at levels dominated by the subject's level, and all of the w-scope dominates the subjects clearance. In the middle example, role R2, of Figure 1a, any subject in one of the levels between that where r-level falls, going up to the level where w-level falls, can be assigned to this role, without violating MAC security properties.

Roles could be constructed which cannot be assigned to untrusted subjects, because doing so would allow such subjects to be able to either read up or write down. Some examples of this are shown in Figure 1b. If the r-level of the role is strictly greater than the w-level of the role (Figure 1b, all three examples), then a subject assigned to this role at a clearance level strictly less than r-level would be able to read up, and a subject at a clearance level strictly greater than w-level would be able to write down. Such a role then must either not exist in the role graph, or must never be assigned to untrusted subjects.

Role R6 in Figure 1b shows that the overlap of the r-scope and w-scope must not be more than one level. An untrusted subject whose clearance falls into a security level that lies strictly between w-level (on the bottom) and r-level (at the top), would be able to both read up and write down.

For the kinds of roles shown in Figure 1b, **trusted** subjects could be assigned to these roles if their clearances satisfy the simple security property of MAC with respect to the r-level of the role.

All of this discussion leads us to the following constraints on roles with non-empty w-scopes:

> **Constraint 2:** An untrusted subject s may be assigned to a role R, only if all of the following hold:
>
> 1. the w-level of R must dominate the r-level of R, and
> 2. $\lambda(s) \geq r$-level of R (i.e. Constraint 1 holds), and
> 3. $\lambda(s) \leq w$-level of R.

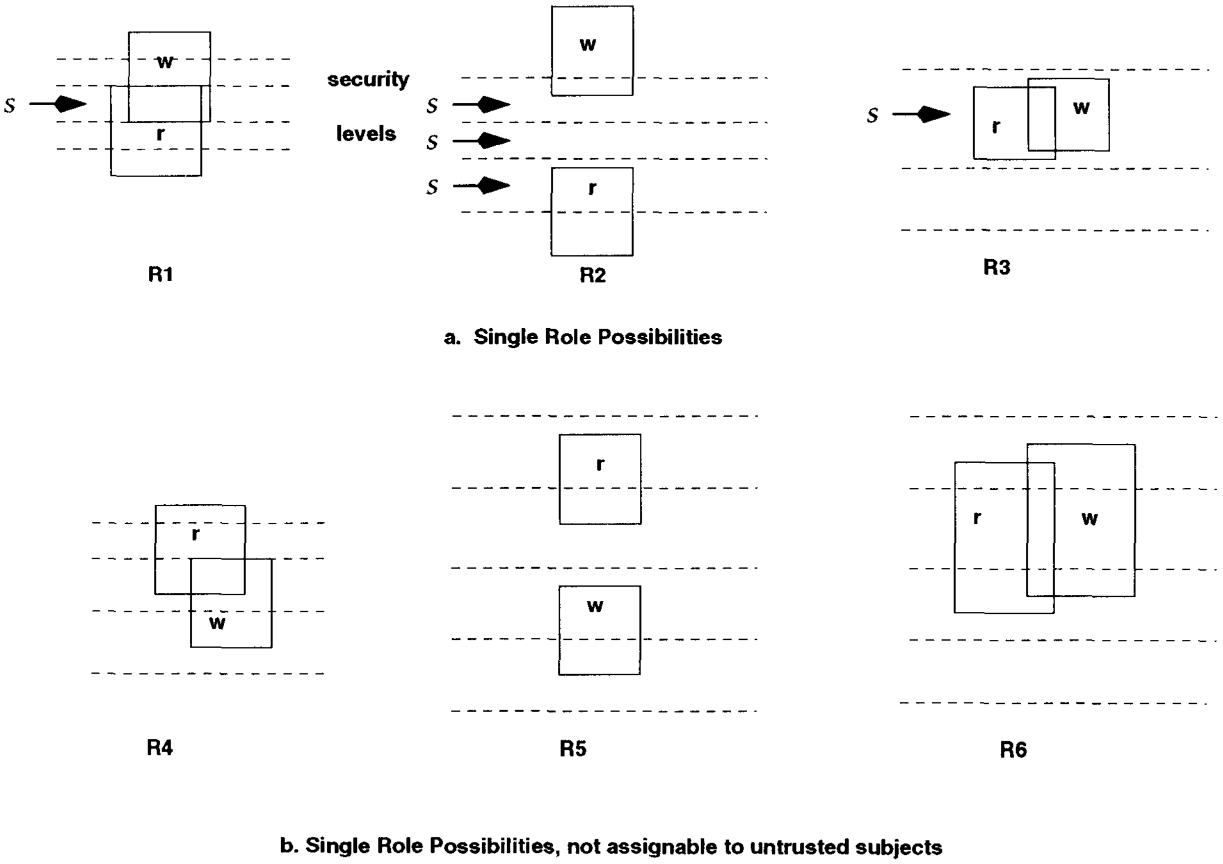
We can see from the examples in the figures that the first condition eliminates all of the cases in Figure 1b. For role R6 it is also impossible to satisfy points 2 and 3 simultaneously.

Points 2 and 3 give us the following:

> **Role Lemma:** A role R is assignable to an untrusted subject according to Constraint 2 only if
>
> w-level of $R \geq r$-level of R

We can also see that these conditions greatly restrict the structure of a single role with respect to security levels. We will see below that the combination of these constraints with role graph edges makes the possible role graph structures even more restricted.

Note also that if the modified $\star$-property is being observed (which, as defined in [6], says that an untrusted subject s can write to object o only if $\lambda(s) = \lambda(o)$), then the entire w-scope of a role must lie within one security classification for it to be assignable.



a. Single Role Possibilities

b. Single Role Possibilities, not assignable to untrusted subjects

Figure 1: Single Roles

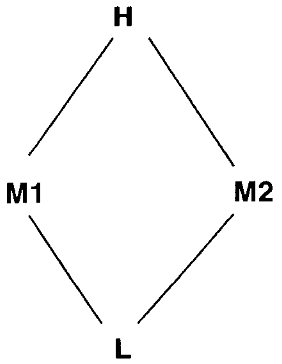


Figure 2: Security Lattice

4.2 Single Roles with a Security Lattice

Consider the example security lattice in Figure 2. Suppose a read-only role has objects labelled at both M1 and M2. Any subject assigned to such a role must have clearance H, in order to be able to exercise all the privileges in the role.

Similarly, suppose a role has in its w-scope objects labelled at levels M1 and M2. To be able to write all of these objects, a subject must have clearance L.

If a role had its r-scope all labelled M1, and its w-scope all labeled M2, this role would be unassignable, because a subject having clearance H would be able to write down, and a subject having clearance L would be able to read up.

Re-examining the definitions and constraints above, we need to redefine the r-level and w-level as follows:

> The *r-level* of a role is the **least upper bound** of the security levels of the objects in the role's r-scope.
>
> The *w-level*of a role is the **greatest lower bound** of the security levels of the objects in the role's w-scope.

Because we have a lattice, the least upper bound and greatest lower bound exist. With these new definitions, Constraints 1 and 2 hold. Note that in the case of Constraint 2, there are many scenarios in which a $\lambda(s)$ satisfing both points 2 and 3 does not exist.

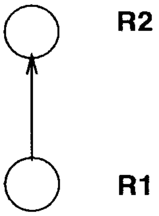


Figure 3: A Single Edge

We will use these revised definitions of r-level and w-level in the rest of the paper.

4.3 Role Graph Edges

Consider two roles joined by an edge, R1 → R2, (Figure 3). By the structure of the role graph, all the privileges of R1 are available to subjects assigned to R2. Consider first roles that are read-only. All of R1's r-scope is contained in R2's r-scope, by definition of the role graph. By definition of r-level, the r-level of R2 is the least upper bound of r-level of R1 and all the objects locally in R2's r-scope. So there could be a chain of these read only roles in the graph, which might have increasing r-levels as one goes in the senior direction in the role graph, or if there is a high (security label) piece of data in R1's r-scope, all the roles in a chain of roles senior to R1 will have the high label.

This gives us the following result for edges between read-only roles. By the definitions of r-level and the role graph, it will always hold. As long as Constraint 1 holds when subject-role assignments are made, edges between read-only nodes cannot create nodes that are unassignable to untrusted subjects.

> **Edge Lemma 1:** If there is an edge R1 → R2 in a role graph such that both R1 and R2 have empty w-scope, then
>
> the r-level of R2 $\geq$ r-level of R1.

Let us consider now the case where there is a role graph edge R1 → R2, such that both roles have nonempty w-scope, and assume that R2 has objects in its w-scope that are not in R1's w-scope, and these objects are at a security classification that is greater than any object in R1's w-scope. By definition of w-level, the w-level of R2 will equal that of R1. If these new objects in R2's w-scope are at a security classification that is incomparable with w-level of R1, then the w-level of R2 will be the glb of all these security classifications. If the nodes are write-only, this is not a problem. However, if there are objects in the r-scopes, it is possible that the unassignable roles like those shown in Figure 1 will be created.

> **Edge Lemma 2:** If there is an edge R1 → R2 in a role graph such that both R1 and R2 have non-empty w-scope, then:
>
> w-level of R2 = glb(w-level of R1, w-level of (w-scope(R2) - w-scope(R1))

Note that if w-scope of R2 = w-scope of R1, then their w-levels are always equal (this is the case where no new privileges with write operations are local to R2).

There are 3 possible states for a role: read-only, write-only or read-write. This gives us 9 possible combinations for an edge R1 → R2. Some of them are impossible, because all privileges of R1 must be inherited by R2, by the structure of the role graph. Table 1 enumerates the possibilities and the constraints that apply. Constraints 1 and 2 determine whether or not a given role is assignable to a subject.

Consider the implications of these two lemmas by following paths in a role graph from MinRole to MaxRole. Some examples are shown in Figures 4 and 5. By the Role Lemma, roles are assignable to untrusted subjects only if their w-level dominates their r-level. This can be achieved in several ways. One can have a chain of read-only roles whose r-level is non-decreasing as we move along a path from MinRole to MaxRole. Similarly, a chain of write-only roles with non-increasing w-level is possible. When there are read-write roles, the interaction of Edge Lemma 2 with Edge Lemma 1 means that if one follows a path in a role graph from MinRole to MaxRole, and there is a read-write role with r-level = w-level, this role is assignable to subjects at this level. Objects can be added to w-scope with higher classifications without altering w-level. As soon as any objects are added to a role with higher r-level, the role becomes unassignable.

R1	R2	
r-only	r-only	Edge Lemma 1
r-only	w-only	not possible
r-only	read-write	Edge Lemma 1
w-only	r-only	not possible
w-only	w-only	Edge Lemma 2
w-only	read-write	Edge Lemma 2
read-write	r-only	not possible
read-write	w-only	not possible
read-write	read-write	Edge Lemma 1 ∧ Edge Lemma 2

Table 1: Possible edges R1 → R2 in a role graph and the MAC security implications

An example role graph with security labels indicated is shown in Figure 4. We are assuming for this example that the security levels are Top Secret > Secret > Unclassified, i.e. that they are linearly ordered. Roles are shown by giving their r-level and/or w-level (e.g. r-u means the r-level is Unclassified, w-s means the w-level is Secret, etc. A missing r- or w-level means it is a write-only or read-only role.) Regions of the graph assignable to subjects at different clearances are indicated. As one follows a path from MinRole to MaxRole, once one enters a role with non-empty w-scope, all remaining roles on this path must be at the same security level, or one enters the region of the graph that cannot be assigned to untrusted users.

If the security lattice contains incomparable levels, as in Figure 5, then a node like (w-u, r-u) which has (w-s2) and (w-s1, r-u) as its junior nodes is actually assignable at the Unclassified security level.

5 Conclusions

The structure of the role graphs that have assignable roles are very restricted compared to

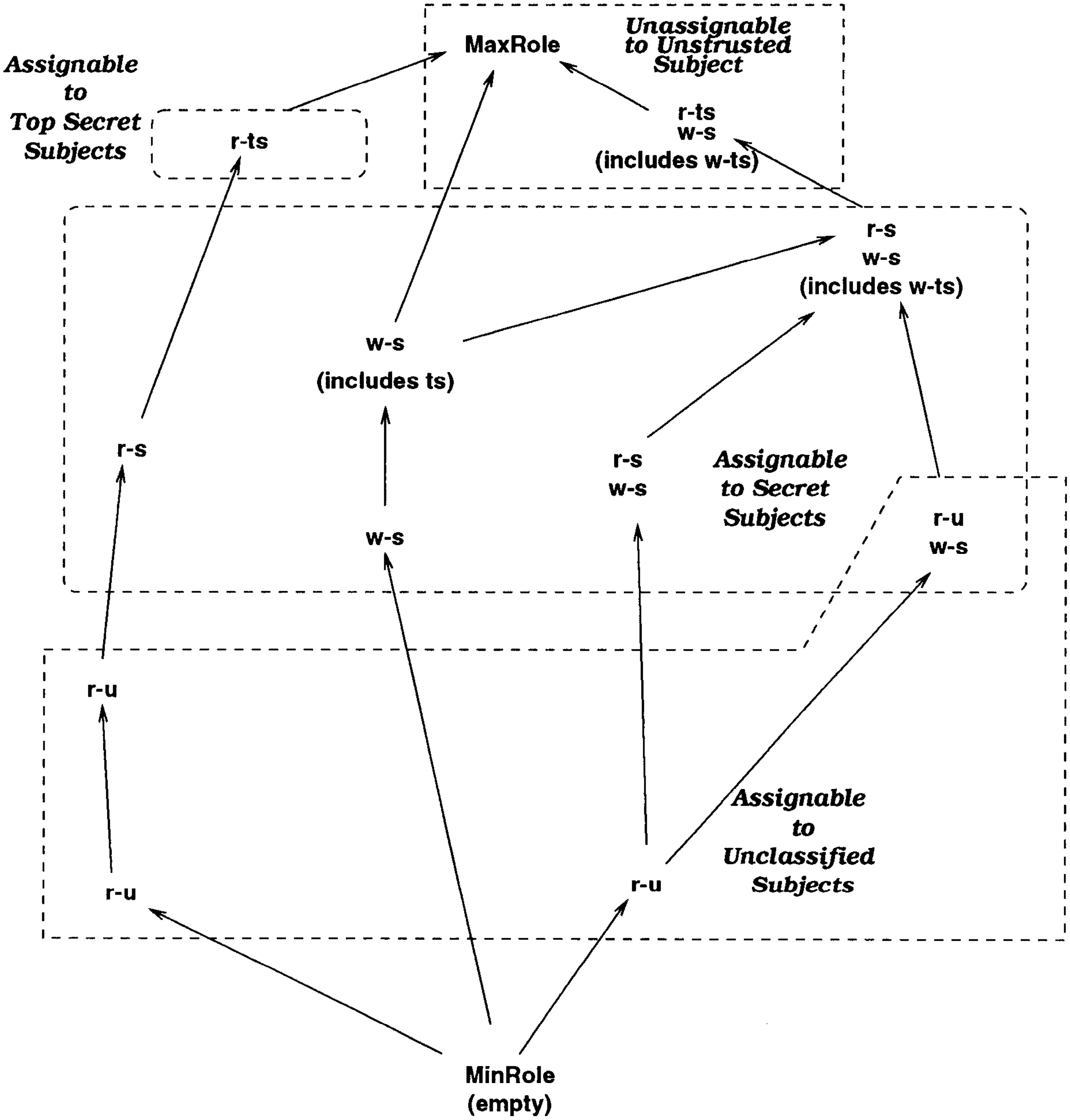

MaxRole
Unassignable to Unstrusted Subject
r-ts w-s (includes w-ts)
Assignable to Top Secret Subjects
r-ts
r-s w-s (includes w-ts)
w-s (includes ts)
r-s
r-s w-s
Assignable to Secret Subjects
w-s
r-u w-s
r-u
r-u
r-u
Assignable to Unclassified Subjects
MinRole (empty)

Figure 4: Possible Role Graph Structures

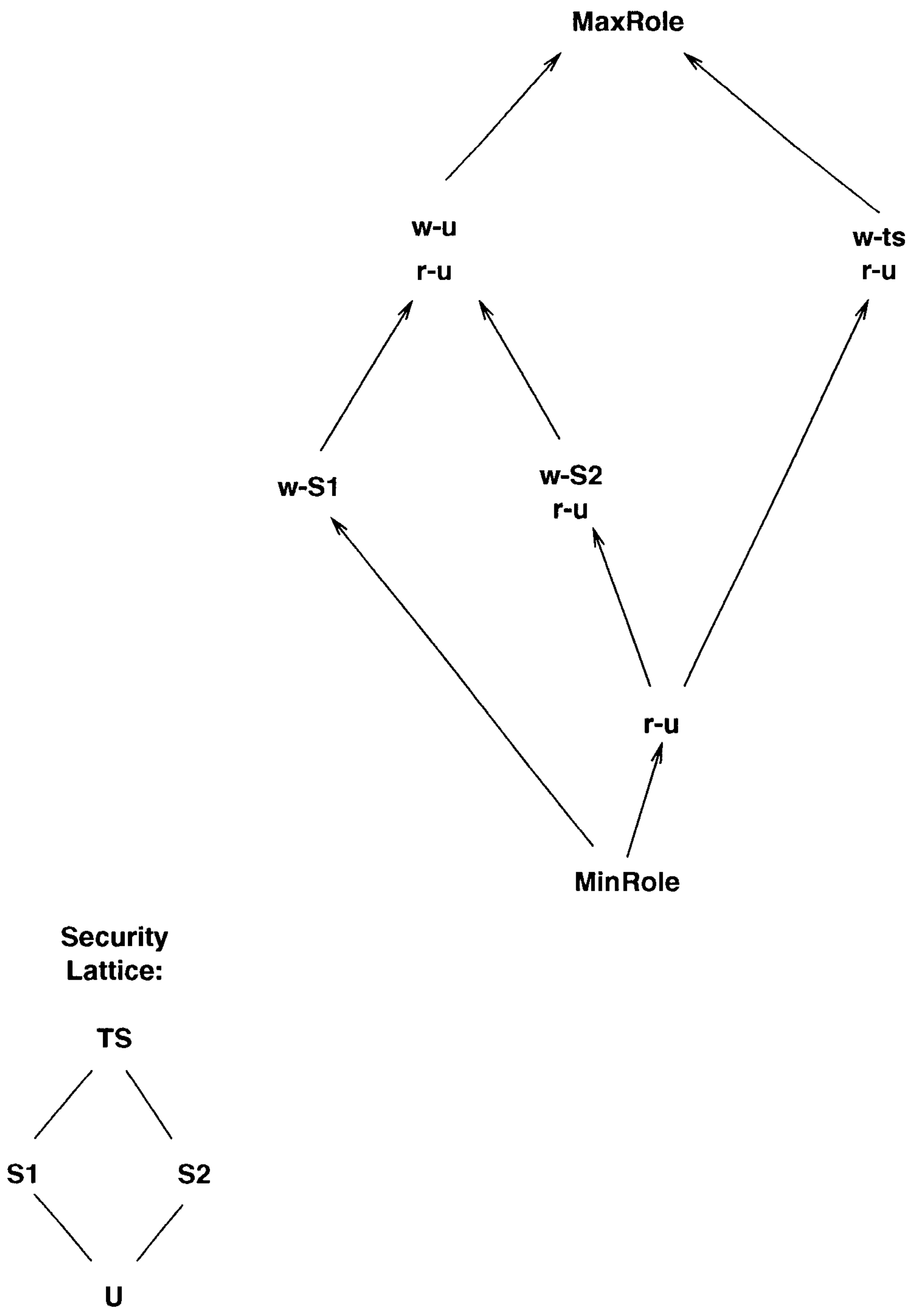


Figure 5: A role graph with all but MaxRole assignable

general role graphs. In a normal role graph, we do not expect to have many read-only or write-only roles. Yet it is really only within the read-only roles that we get much differentiation in security levels.

It is possible to modify the role graph algorithms so that privileges are analyzed to produce the *r*-scope and *w*-scope for each role. Security classifications must then be associated with each object. Then the role graph algorithms can be modified so that they only allow assignable roles or at least raise a warning when roles unassignable to untrusted subjects are created.

In other words, it is possible to analyze every role and every edge in a role graph to see if the roles are assignable, and at what levels they are assignable. All subject-role assignments must adhere to Constraint 1. Untrusted subject-role assignments must adhere to Constraints 2. Roles assignable to untrusted subjects must either be read-only, write-only or follow the restrictions of the Role Lemma.

Acknowledgements

This research was supported by the Natural Sciences and Engineering Research Council of Canada.

References

[1] D.E. Bell and L.J. La Padula. Secure Computer System: Unified Exposition & Multics Interpretation. Technical report, Technical Report MTIS AD-A023588, MITRE Corporation, 1975.

[2] S. Castano, M. Fugini, G. Martella, and P. Samarati. *Database Security.* Addison-Wesley, 1994.

[3] M. Nyanchama. *Commercial Integrity, Roles and Object Orientation.* PhD thesis, Department of Computer Science, The University of Western Ontario, London, Canada, Sept. 1994.

[4] M. Nyanchama and S. L. Osborn. Access rights administration in role-based security systems. In J. Biskup, M. Morgenstern, and C. E. Landwehr, editors, *Database Security, VIII, Status and Prospects, Proceedings of the IFIP WG11.3 Working Conference on Database Security*, pages 37–56. North-Holland, 1994.

[5] M. Nyanchama and S. L. Osborn. Modeling mandatory access control in role-based security systems. In D.L. Spooner, S.A. Demurjian, and J.E. Dobson, editors, *Proceedings of the IFIP WG 11.3 Ninth Annual Working Conference on Database Security*, pages 129–144. Chapman & Hall, 1995.

[6] R.S. Sandhu. Lattice-based access control models. *Computer*, 26:9–19, Nov. 1993.

[7] R.S. Sandhu. Role hierarchies and constraints for lattice-based access controls. In *Computer Security - ESORICS 96*, pages 65–79. Springer Verlag, 1996. Lecture Notes 1146.

[8] R.S. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman. Role-based access control models. *Computer*, 29:38–47, Feb. 1996.

[9] R.S. Sandhu and C. Youman, editors. *First ACM Workshop on Role-Based Access Control.* Association for Computing Machinery, Nov. 30-Dec. 1 1995.