Conflict of interest policies in role-based access control

Jason Crampton* and George Loizou

Department of Computer Science, Birkbeck College, University of London,

Malet Street, London, WC1E 7HX, England

e-mail: ccram01@dcs.bbk.ac.uk

November 14, 2000

Abstract

Although various schemes have been developed for expressing separation of duty constraints and policies, the semantics have usually been taken for granted. In this paper we introduce a simple but general approach to policy specification with a well-defined semantics, and show that in particular the approach can be applied to separation of duty policies in role-based access control. We demonstrate that the range of policies that can be considered is thereby extended and show that these policies increase the expressive power of the role-based access control model. We also briefly discuss a systematic way of combining two or more separation of duty policies and additional technical properties of our approach.

1 Introduction

Access control models and policies originally concentrated on preserving the integrity and confidentiality of information in a computer system [3, 4, 12]. One can regard such policies

*Supported by EPSRC Award 98317878

in two complementary ways. Either one specifies what constitutes a violation of confidentiality (or integrity), or one specifies what constitutes satisfaction of confidentiality (or integrity).

To make this explicit, let us consider such policies in the context of the Harrison-Ruzzo-Ullman model [12]. We assume the existence of a set of subjects, S, a set of objects, O, and a set of access rights (or modes), AR. A policy, $P^- \subseteq O \times S \times AR$, specifies which triples are to be prohibited. An access control monitor which implements P^- would grant subject s access right r to object o only if $(o, s, r) \notin P^-$. The complementary view is that the policy $P^+ \subseteq O \times S \times AR$ specifies which triples are permitted. Subject s is granted access right r to object o only if $(o, s, r) \in P^+$.

In recent years separation of duty policies have attracted considerable interest, particularly in the role-based access control community. In order to discuss the historical background and motivation of this work we introduce the following temporary definition with respect to a role-based access control model (RBAC96 [19], for example). This definition will be generalised in Section 3.

Definition 1.1 *A* separation of duty constraint *is a subset of R, the set of roles. A* separation of duty policy *is a set of separation of duty constraints.*

Intuitively, a separation of duty constraint specifies which roles should not "co-exist" in some context dependent sense. For example, the accepted interpretation of static separation of duty requires that no two roles in the same constraint be assigned to the same user, while dynamic separation of duty requires that no two roles in the same constraint be assigned to the same session [11]. In the ensuing discussion we will only consider static separation of duty, with the understanding that identical arguments, modulo implementation, can be applied to dynamic separation of duty.

There have been notable advances in the understanding and articulation of separation of duty constraints in recent years, in particular the NIST model and its implementation [11] and the RCL 2000 language [2] (a refinement of the RSL99 language [1]).

The NIST Model A static separation of duty policy is modelled and implemented as a symmetric, anti-reflexive relation $ssd \subseteq R \times R$. The interpretation of the policy, naturally enough, is that for each constraint, $(r_1, r_2) \in ssd$, the access control system (Admin Tool) should prohibit any actions which result in both r_1 and r_2 being assigned to a user. In a similar style to the seminal Bell-LaPadula paper [3], the authors present rules to maintain the integrity of the RBAC Database.

RCL 2000 RCL 2000 is a role-based authorization constraints specification language. The language includes a conflicting role set, $\texttt{CR} = \{R_1, \ldots, R_n\}$, where $R_i \subseteq R$, $1 \leqslant i \leqslant n$. In other words `CR` is a set of constraints and hence, in our terminology, is a separation of duty policy. The language contains RCL 2000 expressions of the form [2]

$$|\texttt{roles*(OE(U))} \cap \texttt{OE(CR)}| \leqslant 1. \tag{1}$$

The semantics of RCL 2000 expressions can be obtained by translating into RFOPL (restricted first order predicate logic). In the case of (1) the interpretation is that no user can be assigned two or more roles which belong to a separation of duty constraint (that is, a member of `CR`).

The language also incorporates a conflicting permissions set, `CP`, and a conflicting user set, `CU`; these sets permit the specification of previously unrecognised and higher assurance separation of duty policies.

Semantics We note the correspondence in operational semantics between the two approaches above. Specifically, both the NIST model and the RCL 2000 expression (1) consider a policy to be violated if two (or more, in the case of RCL 2000) roles in a separation of duty constraint are assigned to a user. It can be seen that a different RCL 2000 expression can be used to change the violation condition of a separation of duty policy,

namely

$$|\texttt{roles*(OE(U))} \cap \texttt{OE(CR)}| < |\texttt{OE(CR)}|. \tag{2}$$

The interpretation of (2) is that the policy `CR` is violated only if all roles in a separation of duty constraint are assigned to a user.

Role exclusion It has been noted that in certain situations it is required that a role have no users assigned to it. One such situation occurs in the Role Graph model in which there is a `MaxRole` senior to all other roles, in the presence of separation of duty constraints [14, 15]. It must certainly be the case that no user be assigned to the `MaxRole` because that user would then be (implicitly) assigned two (or more) roles which are in a separation of duty constraint. Similarly, "de-activated" roles which are due for pruning from the role hierarchy [21] should not have users assigned to them. Clearly, in these situations, roles could be flagged to indicate that no users should be assigned to them.

However, if we wanted to be more sophisticated and exclude certain users from certain roles - any user who is a finance clerk should not be assigned to the role `FD` (finance director), say - then there is little or no work to suggest how this sort of requirement might be articulated and implemented. We will show that such requirements can be expressed naturally within our framework. (It can, in fact, be accomplished in RCL 2000, but it is not identified as a useful part of the language. A similar effect can probably also be achieved in URA97 [18] through careful choices of constraints in the `can-assign` predicate.)

Contribution Our recent work has considered subsets of the powerset of a partially ordered set and their application in access control modelling [6, 9]. This arose from our interest in categorising access control policies [17]. As a result of this work we have formulated a general mathematical characterisation of separation of duty constraints and policies [7].

Separation of duty constraints within the role-based access control model can be treated

as a special case of this characterisation. In this paper we demonstrate how the NIST and RCL 2000 approaches can be accommodated, and how role exclusion constraints can be expressed, giving a novel application of these ideas.

We show that our model combines the simplicity of the NIST approach with the expressive power of the RCL 2000 language. Furthermore, we show it is possible to rigorously define the composition of two separation of duty policies and that all separation of duty policies have a unique *canonical* representation. In its most general form, our approach is expensive to implement. We show that this overhead can be reduced by implementing a simpler policy which is at least as secure as the original policy. This reduction suggests that some RCL 2000 expressions can also be simplified. Finally, we define the *length* of a separation of duty policy and briefly discuss some consequences of this definition.

Structure of the paper In the next section we introduce some basic definitions from the theory of partially ordered sets. In Section 3 we describe our model of separation of duty policies and explain why we use the terminology "conflict of interest" policy. In Section 4 we apply our approach to the particular case of the role-based access control model. In Section 5 we briefly discuss the theoretical development of our model.

2 Posets

Definition 2.1 *A pair* $\langle P, \leqslant \rangle$ *is a* partially ordered set *or* poset *if for all* $p, q, r \in P$,

- $p \leqslant p$,
- $p \leqslant q$ *and* $q \leqslant p$ *implies* $p = q$,
- $p \leqslant q$ *and* $q \leqslant r$ *implies* $p \leqslant r$.

In other words $\leqslant$ is a binary relation on P which is reflexive, anti-symmetric and transitive, respectively. We will write

- $p < q$ if, and only if, $p \leqslant q$ and $p \neq q$;

- $p \parallel q$ if, and only if, $p \not\leqslant q$ and $p \not\geqslant q$.

Hereafter we will write "a poset P" to mean the pair $\langle P, \leqslant \rangle$. In other words, unless explicitly stated, P is assumed to have the ordering $\leqslant$.

Definition 2.2 *Given a poset P, $Q \subseteq P$ is a* chain *if for all $q_1, q_2 \in Q$ either $q_1 \leqslant q_2$ or $q_2 \leqslant q_1$. Q is an* antichain *if for all $q_1, q_2 \in Q$, $q_1 \parallel q_2$. We denote the set of antichains by $\mathcal{A}(P)$.*

Definition 2.3 *Given a poset P and $Q \subseteq P$, we say $q \in Q$ is a* minimal *element if for all $q' \in Q$, $q' \leqslant q$ implies $q = q'$. We denote the set of minimal elements in Q by $\underline{Q}$.*

Lemma 2.1 *For all $Q \subseteq P$, $q \in Q$,*

$$\underline{Q} \subseteq Q, \tag{3}$$

$$\text{there exists } q' \in \underline{Q} \text{ such that } q' \leqslant q, \tag{4}$$

$$\underline{Q} \in \mathcal{A}(P), \tag{5}$$

$$\underline{Q} \text{ is unique.} \tag{6}$$

Proof: The proof is trivial, following immediately from Definition 2.3, and is left as an exercise for the interested reader. ■

3 Conflict of interest policies

Let X be some set of access control artefacts. We will refer to X as an *access control context* (or simply *context*). For example, X may be the set of all roles in a role-based access control model.

An *access control environment* (or simply *environment*), E, is a subset of X. The environment models the relevant access control system data structure. An example of E would be the set of roles assigned to a given user.

Definition 3.1 *A* conflict of interest constraint *(or simply* constraint*) is defined to be a subset of X. A* conflict of interest policy *is defined to be a set of conflict of interest constraints.*

An environment, E, satisfies a conflict of interest policy, α, if, and only if, for all $A \in \alpha$, $A \cap E \subset A$. We denote the set of environments which satisfy α by $\mathcal{E}(\alpha)$. (We also say that α is violated *by E if there exists $A \in \alpha$ such that $A \subseteq E$.)*

In other words, a conflict of interest policy states which subsets of X cannot be present simultaneously in the environment, and is satisfied provided the environment does not include any conflict of interest constraint in the policy. We make the following observations about this definition.

- A singleton set $\{a\} \in \alpha$, $a \in X$ implies that a is prohibited from ever entering the environment E. In other words, policies which address confidentiality or integrity considerations by prohibiting a list of triples are a special case of our framework. Specifically, the policy

  $$P^- = \{x_1, \ldots, x_n\}$$

  can be expressed as the conflict of interest policy

  $$\alpha = \{\{x_1\}, \ldots, \{x_n\}\}.$$

  It is because our framework can accommodate policies which articulate confidentiality and integrity constraints, as well as separation of duty constraints that we prefer the terminology "conflict of interest" rather than separation of duty policies. In this sense, conflict of interest policy means a policy which conflicts with the interest of the system.

- If $\alpha = \{\emptyset\}$ then no environment satisfies α (since $\emptyset \subseteq E$ for all $E \subseteq X$).

- If $\alpha = \emptyset$ then every environment satisfies α (since α contains no constraints).

Let $X = \{1, 2, 3\}$. Table 1 shows three policies $\alpha_1 = \{\{1, 2\}, \{2, 3\}\}$, $\alpha_2 = \{\{1\}, \{2, 3\}\}$, $\alpha_3 = \{\{1\}, \{1, 2\}, \{2, 3\}\}$ and the environments that satisfy (ticked) and violate (crossed) each policy. These policies could be regarded as defined on the subscripts of the set of roles $\{r_1, \ldots, r_n\}$. For example in α_1, the roles r_2 and r_3 form a conflict of interest constraint.

Environment	$\alpha_1 = \{\{1, 2\}, \{2, 3\}\}$	$\alpha_2 = \{\{1\}, \{2, 3\}\}$	$\alpha_3 = \{\{1\}, \{1, 2\}, \{2, 3\}\}$
$\emptyset$	✓	✓	✓
$\{1\}$	✓	✗	✗
$\{2\}$	✓	✓	✓
$\{3\}$	✓	✓	✓
$\{1, 2\}$	✗	✗	✗
$\{1, 3\}$	✓	✗	✗
$\{2, 3\}$	✗	✗	✗
$\{1, 2, 3\}$	✗	✗	✗

Table 1: A comparison of conflict of interest policies and environments

Definition 3.2 *Given two conflict of interest policies, α, β, we say α is* weaker than *(or* less restrictive than *or* is enforced by*) β if $\mathcal{E}(\alpha) \supset \mathcal{E}(\beta)$. Analogously, we say β is* stronger *(or* less restrictive than *or* enforces*) α; α and β are* equivalent *if $\mathcal{E}(\alpha) = \mathcal{E}(\beta)$.*

In Table 1, α_1 is weaker than α_2, for example. From Table 1 we also see that α_2 and α_3 are equivalent. In fact, we have the following result [7].

Proposition 3.1 *Suppose $\alpha \in \mathcal{P}(\mathcal{P}(X))$ and $A \subset B$ for some $A, B \in \alpha$. Define $\alpha' = \alpha \setminus \{B\}$. Then an environment, E, satisfies α if, and only if, E satisfies α'.*

This leads naturally to the definition of a *canonical* representation of a conflict of interest policy.

Definition 3.3 *Given a conflict of interest policy $\alpha \in \mathcal{P}(\mathcal{P}(X))$, the* canonical representation *of α is defined to be $\underline{\alpha} \in \mathcal{A}(\mathcal{P}(X))$ - the set of minimal elements in α.*

The canonical representation of a policy is unique by (6), and is equivalent to the original policy by Proposition 3.1. For example α_2 is the canonical representation of α_3 in the example given in Table 1.

Note that in Definition 3.1 we assumed nothing about the set X. If, in fact, the context supports some sort of inheritance, that is $\langle X, \leqslant \rangle$ is a partially ordered set, then we observe that in general a conflict of interest constraint should be defined to be an antichain in X rather than a subset of X.

Consider the role hierarchy in Figure 1 and the policy $\alpha = \{\{r_1, r_3\}, \{r_2, r_3\}\}$. It is clear that if r_1 enters the environment, then so do r_2 and r_3, violating both constraints. Hence the policy α can be reduced to the policy $\alpha' = \{\{r_1\}\}$.

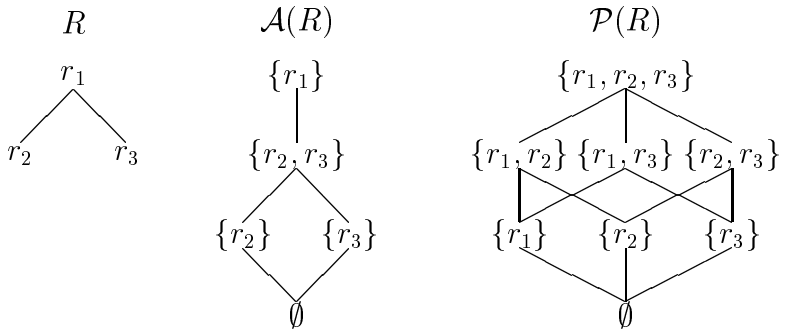

Figure 1: A simple role hierarchy, R, $\mathcal{A}(R)$ and $\mathcal{P}(R)$

In general, therefore, a conflict of interest policy in a role-based access control model is a member of $\mathcal{A}(\mathcal{A}(R))$. In other words, the constraints of a conflict of interest policy are elements of $\mathcal{A}(R)$ rather than $\mathcal{P}(R)$; see, for example, Figure 1. We observe that in the specification of RCL 2000 it is not mentioned whether the elements of `CR` should be antichains or not. (In the case of an unordered set X - that is, the order relation is the empty set - the set of antichains is simply $\mathcal{P}(X)$.)

4 Conflict of interest policies in role-based access control

We now consider the interpretation of our approach in the context of a role-based access control model. We will assume the reader is familiar with the RBAC96 family of models [19]. We will compare the NIST approach and the RCL 2000 language with our own approach and demonstrate that we can accommodate both.

We first consider the simplest context, namely $X = R$. Recall that the NIST approach was to represent a separation of duty policy as a (binary) symmetric, anti-reflexive relation, ssd. In other words, a NIST policy, α_{NIST}, is simply a set of pairs of roles. Hence we can represent α_{NIST} by

$$\alpha_{\text{NIST}} = \{\{r_{1_1}, r_{1_2}\}, \ldots, \{r_{n_1}, r_{n_2}\}\}, \tag{7}$$

where $\{r_{i_1}, r_{i_2}\} \in \alpha_{\text{NIST}}$ if, and only if, $(r_{i_1}, r_{i_2}) \in ssd$ for all $1 \leqslant i \leqslant n$. The NIST model requires that $r_{i_1} \parallel r_{i_2}$ for all $1 \leqslant i \leqslant n$. Hence each constraint in α_{NIST} is an antichain.

The RCL 2000 expression (1) is equivalent to the policy α_{RCL}, where

$$\alpha_{\text{RCL}} = \{\{r_{i_j}, r_{i_k}\} : \ 1 \leqslant j < k \leqslant |R_i|, \ 1 \leqslant i \leqslant n\}. \tag{8}$$

To see this, we observe that (1) merely expresses the conditions for violation of the conflict of interest policy `CR`. Specifically `CR` is violated if two or more roles in a conflict of interest constraint are assigned to a user. In our formulation this is simply achieved by constructing a policy of all possible pairs of conflicting roles. Furthermore, the RCL 2000 expression (2) is equivalent to the policy

$$\alpha_{\text{RCL}'} = \{R_1, \ldots, R_n\}.$$

Our approach is more flexible since we can easily accommodate conflict of interest con-

straints comprising different numbers of roles, whereas in RCL 2000, we believe a separate expression (defining policy violation conditions) will need to be written for each element of `CR`.

Note that we can also state global role exclusion policies - $\alpha = \{\{\texttt{MaxRole}\}\}$ - for example.

We next consider the context $X = U \times R$. This expands the range of policies enormously. In this case, the appropriate environment is the user-role assignment relation [19]; we consider the following simple examples.

- $\alpha_1 = \{\{u, r_1\}, \ldots, \{u, r_n\}\}$ is a role exclusion policy stating that user u cannot occupy any of the roles $r_1, \ldots, r_n$.

  We note the following useful application of such a policy. We recall that the role-based access control model is policy neutral [19], and that it is of considerable value to demonstrate that such a model can be used to simulate mandatory and discretionary access control models [14, 20, 16]. It has been convincingly shown that role-based access control can indeed simulate mandatory access control [16] by considering the security lattice, L, as two distinct read and write role hierarchies L_R and L_W, respectively, where L_R is *isomorphic* to L and L_W is the *dual* of L_R [10].

  However, we believe the constraints introduced in [16] to enforce the information flow policy that is an integral part of the mandatory access control model are rather complicated. We suggest that to achieve this we can simply define a role exclusion policy of the form α_1 for each user u, where $\{r_1, \ldots, r_n\}$ is an antichain in L. Figure 2 shows a security lattice for the security labels

  $$\texttt{unclassified} < \texttt{classified} < \texttt{secret} < \texttt{top secret}$$

  which we will abbreviate to `u`, `c`, `s`, and `t`, respectively, and two (needs-to-know) categories [3], `a` and `b`. If a user, u, has security clearance `ca`, the conflict of interest

policy

$$\{\{(u, \mathtt{sa})\}, \{(u, \mathtt{cb})\}\}$$

preserves the information flow policy defined by the lattice by preventing u being assigned to, and hence activating, any roles other than `u` and `ca`. (Of course, in a role-based access control implementation there would actually be a read and a write lattice, but the example policy can be extended in the obvious way to accommodate this.)

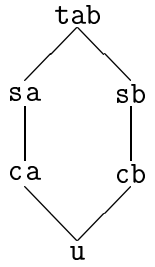

Figure 2: A security lattice

- $\alpha_2 = \{\{(u_1, r_1), (u_2, r_2)\}, \{(u_1, r_2), (u_2, r_1)\}, \{(u_1, r_1), (u_1, r_2)\}, \{(u_2, r_1), (u_2, r_2)\}\}$ is a conflicting user and conflict of interest policy in which the users u_1 and u_2 cannot be assigned both r_1 and r_2 either as individual users (the third and fourth constraints in α_2) or one role to each of the users (the first and second constraints in α_2). (The intuition here, as first identified in [1], is that there may be sensitive combinations of users - family members, say - which should be prevented from occupying sensitive combinations of roles.)

We note that by changing the context, X, we can formulate policies about sensitive combinations of permissions, or of combinations of permissions and roles, etc. We also note the possibility of "telescoping" the notation so that a policy of the form $\alpha = \{V \times S\}$, where $V \subseteq U$ and $S \subseteq R$, is equivalent to $\{\{(u, r), (u', r')\} : u, u' \in U,\ r, r' \in R\}$. In particular,

$\alpha_2 \equiv \{\{u_1, u_2\} \times \{r_1, r_2\}\}$. We can also, for example, specify a role exclusion policy for a group of users, $V \subseteq U$, by $\{V \times \{r\}\} = \{\{u, r\} : u \in V\}$.

In conclusion, we observe that in the role-based access control context, our approach is essentially an extension of the NIST model to include arbitrary constraints and more complicated contexts, and that by deriving equivalent policies to the examples of RCL 2000 expressions in [2] we believe our approach has similar expressive power. It is beyond the scope of this paper to prove that RCL 2000 and our approach are equivalent.

In [7] we demonstrate the generality of approach by showing that we can form conflict of interest policies in the protection matrix model [13], in which the context is either $S \times O$ or $S \times O \times AR$ depending on the granularity of conflict of interest policy required.

5 Further properties of conflict of interest policies

We now consider how we might compare and compose two conflict of interest policies. We first present a motivating example.

Example 5.1 *Let $X = \{1, 2, 3\}$ and suppose we have two conflict of interest policies*

$$\alpha = \big\{\{1\}, \{2, 3\}\big\} \quad \textit{and} \quad \beta = \big\{\{2\}, \{1, 3\}\big\}.$$

Then the only environments which satisfy both α and β are $E = \emptyset$ and $E = \{3\}$.

We want to define an operation $\circ$ on conflict of interest policies such that $\alpha \circ \beta$ enforces both α and β, but is no more restrictive than necessary. For example, if we were to define $\circ$ such that $\alpha \circ \beta = \{\{1\}, \{2\}, \{3\}\}$, then this operation is too strong as the only environment which satisfies $\alpha \circ \beta$ is $\emptyset$.

That is, we wish to find an ordering on the set of conflict of interest policies and define $\alpha \circ \beta$ to be the greatest lower bound of α and β.

Definition 5.1 *Let $\langle X, \leqslant \rangle$ be a poset. Then for all $\alpha, \beta \in \mathcal{A}(X)$, we define*

$$\alpha \preccurlyeq \beta \textit{ if, and only if, for all } b \in \beta, \textit{there exists } a \in \alpha \textit{ such that } a \leqslant b.$$

In [8] we prove that the binary operation

$$\alpha \circ \beta =_{\text{def}} \underline{\alpha \cup \beta}$$

is the greatest lower bound of α and β (with respect to the ordering $\preccurlyeq$). In particular, we can apply Definition 5.1 to the poset $\langle \mathcal{P}(X), \subseteq \rangle$ and hence define an ordering on $\mathcal{A}(\mathcal{P}(X))$. That is, $\alpha \circ \beta$ is the unique policy that incorporates necessary and sufficient information to enforce α and β simultaneously.

Example 5.2 *Consider the policies* $\alpha = \{\{1\}, \{2,3\}\}$ *and* $\beta = \{\{2\}, \{1,3\}\}$ *from Example 5.1. Then*

$$\begin{aligned} \alpha \circ \beta &= \underline{\alpha \cup \beta} \\ &= \underline{\{\{1\}, \{2,3\}, \{2\}, \{1,3\}\}} \\ &= \{\{1\}, \{2\}\} \end{aligned}$$

It can be seen that $\mathcal{E}(\alpha \circ \beta) = \{\emptyset, \{3\}\}$ *as required.*

The following proposition demonstrates that the formal definition of an ordering, $\preccurlyeq$, on the set of conflict of interest policies corresponds exactly to the intuitive definition of strength given in Definition 3.2.

Proposition 5.1 *For all* $\alpha, \beta \in \mathcal{A}(\mathcal{P}(X))$, $\alpha \preccurlyeq \beta$ *if, and only if,* α *is stronger than* β.

Proof: The reader is referred to [7]. ■

In [6, 8] we prove that the set of conflict of interest policies forms a complete lattice, and in [7] we establish an upper bound for $|\mathcal{A}(\mathcal{P}(X))|$, the number of conflict of interest policies, as a function of $|X|$.

Given the large number of conflict of interest policies which can be defined even for small values of $|X|$, the following general principle is important for implementation purposes.

Given an arbitrary conflict of interest policy, $\alpha = \{A_1, \ldots, A_n\}$, where $A_i \subseteq X$ for $1 \leqslant i \leqslant n$, we can construct the policy α' in which every constraint A_i such that $|A_i| > 1$ is replaced by constraints comprising all possible pairs of elements from A_i, and every constraint A_i such that $|A_i| = 1$ is a constraint in α'. The resulting policy, α', is at least as strong as α. Formally, we have the following result.

Proposition 5.2 *Let $\alpha = \{A_1, \ldots, A_n\} \in \mathcal{A}(\mathcal{P}(X))$ be a conflict of interest policy, and define*

$$\alpha' = \{\{a_{i_j}, a_{i_k}\} : 1 \leqslant j < k \leqslant |A_i|, |A_i| > 1, 1 \leqslant i \leqslant n\} \cup \{A_i : |A_i| = 1, 1 \leqslant i \leqslant n\}.$$

Then $\alpha' \preccurlyeq \alpha$.

Proof: The proof is immediate from Definition 5.1 and Proposition 5.1. (Clearly by construction, for all $A_i \in \alpha$, there exists at least one $A' \in \alpha'$ such that $A' \subseteq A_i$, and hence $\alpha' \preccurlyeq \alpha$. That is, α' is stronger than α.) ■

We will denote the set of resulting policies by $\mathcal{D}(\mathcal{P}(X))$ ($\mathcal{D}$ for doubleton). Formally, we define

$$\mathcal{D}(\mathcal{P}(X)) = \{\alpha \in \mathcal{A}(\mathcal{P}(X)) : |A| \leqslant 2, \text{ for all } A \in \alpha\}.$$

In [7] we define

$$l(\alpha) = \sum_{A \in \alpha} |A|$$

to be the *length* of a conflict of interest policy, α, and prove that

$$l(\alpha) \leqslant \lceil n/2 \rceil \binom{n}{\lceil n/2 \rceil},$$

where $n = |X|$, $\lceil n/2 \rceil$ is the smallest integer larger than $n/2$, and $\binom{n}{\lceil n/2 \rceil}$ is the binomial coefficient. Clearly to determine whether the addition of $x \in X$ to the environment E will violate α requires at most $l(\alpha)$ comparisons with E (to ascertain whether x is in a constraint and, if so, whether each of the other elements in a constraint are in E). When n is large and α is an arbitrary element of $\mathcal{A}(\mathcal{P}(X))$, this is likely to be too expensive in a practical system.

However, if $\delta \in \mathcal{D}(\mathcal{P}(X))$, then $l(\delta) \leqslant n(n-1)$, and hence (assuming comparisons with elements of E can be made efficiently, in some sense) we have a manageable worst case for determining whether the addition of an element to E will violate δ.

6 Conclusion

We have sketched our general framework for considering conflict of interest policies and shown that it can be used to consider separation of duty constraints and separation of duty policies in role-based access control. We have demonstrated that it has the economy of the NIST model and the versatility of the RCL 2000 language, that certain useful policies can be easily defined and given an insight into the difficulty of implementing our approach.

We intend to expand our interest in role-based access control to consider the following issues:

- to develop the notation suggested at the conclusion of Section 4 into a formal language which incorporates our formulation of conflict of interest policies;
- to consider further applications of the formal study of antichains to the role-based access control model. For example, our most recent work [5] investigates antichains in user-role assignment;
- to implement our model for conflict of interest policies in a role-based access control system;
- to prove our approach has equivalent expressive power to RCL 2000.

References

[1] G-J. Ahn and R.S. Sandhu. The RSL99 language for role-based separation of duty constraints. In *Proceedings of Fourth ACM Workshop on Role-Based Access Control*, pages 43–54, Fairfax, Virginia, October 1999.

[2] G-J. Ahn and R.S. Sandhu. Role-based authorization constraints specification. *ACM Transactions on Information and System Security*, 3(4), November 2000.

[3] D.E. Bell and L. LaPadula. Secure computer systems: Unified exposition and MULTICS interpretation. Technical Report MTR-2997, Mitre Corporation, March 1976.

[4] K.J. Biba. Integrity considerations for secure computer systems. Technical Report MTR-3153, Mitre Corporation, April 1977.

[5] J. Crampton and G. Loizou. Role-based access control: A constructive appraisal. In preparation.

[6] J. Crampton and G. Loizou. The completion of a poset in a lattice of antichains. *Submitted*, October 2000. Preliminary version available as Technical Report BBKCS-0001.

[7] J. Crampton and G. Loizou. On the structural complexity of conflict of interest policies. To be submitted, November 2000.

[8] J. Crampton and G. Loizou. Two partial orders on the set of antichains. Research note, September 2000.

[9] J. Crampton, G. Loizou, and G. O'Shea. Evaluating access control. *Submitted*, 1999. Preliminary version available as Technical Report BBKCS-9905.

[10] B.A. Davey and H.A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, Cambridge, UK, 1990.

[11] S.I. Gavrila and J.F. Barkley. Formal specification for role based access control user/role and role/role relationship management. In *Proceedings of Third ACM Workshop on Role-Based Access Control*, pages 81–90, Fairfax, Virginia, October 1998.

[12] M.A. Harrison, W.L. Ruzzo, and J.D. Ullman. Protection in operating systems. *Communications of the ACM*, 19(8):461–471, August 1976.

[13] B.W. Lampson. Protection. *ACM Operating Systems Review*, 8:437–443, 1974.

[14] M. Nyanchama and S. Osborn. The role graph model. In *Proceedings of First ACM Workshop on Role-Based Access Control*, pages II25–II31, Gaithersburg, Maryland, October 1995.

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

[16] S. Osborn, R. Sandhu, and Q. Munawer. Configuring role-based access control to enforce mandatory and discretionary access control policies. *ACM Transactions on Information and System Security*, 3(2), May 2000.

[17] G. O'Shea. *Access Control in Operating Systems.* PhD thesis, Birkbeck College, University of London, July 1997.

[18] R.S. Sandhu, V. Bhamidipati, and Q. Munawer. The ARBAC97 model for role-based administration of roles. *ACM Transactions on Information and System Security*, 1(2):105–135, February 1999.

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

[20] R.S. Sandhu and Q. Munawer. How to do discretionary access control using roles. In *Proceedings of Third ACM Workshop on Role-Based Access Control*, pages 47–54, Fairfax, Virginia, October 1998.

[21] R.S. Sandhu and Q. Munawer. The RRA97 model for role-based administration of role hierarchies. In *Proceedings of 14th Annual Computer Security Applications Conference*, Phoenix, Arizona, USA, 1998.