

Authorisation and antichains

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`

January 23, 2001

Abstract

We present a summary of our recent work on partial orders and their application to access control modelling. In particular, we introduce a framework for separation of duty policies and a new access control model. We briefly discuss a special case of this model, HSS RBAC, which is our variation of a role-based access control model.

1 Introduction

Authorisation in computer systems is concerned with the ways in which users (*subjects*) can access *objects* in the computer system: informally, “who can do what?”. The modelling of authorisation (or *access control*) has a long history, from the seminal papers of the early 1970s [2, 3, 13], through to recent developments in role-based access control modelling [18, 19]. Some access control models, such as the Bell-LaPadula model [2], enforce a particular *authorisation policy* (or *access control policy*), while others, like role-based access control models, are “policy neutral” and can be used to implement many types of access control policy.

The work of Greg O’Shea [15] proposed a logical framework for reasoning about the implementation of security in a discretionary access control system. These ideas were further developed in [9, 10]. In particular, in [9] we use this framework to compare the requirements of an abstract specification of security (an access control policy) with the actual security provided by the implementation of the access control policy (as realised through configuration of file permissions and access control lists). (We will refer to that part of a computer system which is concerned with authorisation as the *reference monitor* or *access control sub-system*.)

We express access control policies as Horn clauses - a subset of first-order logic [14]. (This representation was chosen because we were considering UNIX systems and building

a deductive database in Prolog to reason about the security of those systems. Furthermore, the semantics of Prolog's *negation as failure rule* and UNIX authorisations are identical for practical purposes.)

As a result of considering access control policies, we became increasingly interested in articulating separation of duty policies. This led us to investigate Sperner families of sets [4, 20] in power sets and then in general partially ordered sets (posets). As a result we prove in [8] that every poset has a completion in a lattice of antichains. This result has proved to be useful for two reasons. Firstly, it has allowed us to realise our original objective, namely to formulate a general framework for expressing separation of duty policies. Secondly, we observed that the security lattice of the Bell-LaPadula model can be considered as a completion of the set of security labels and needs-to-know categories. This observation has led us to a generalisation of the Bell-LaPadula model - the hierarchical secure systems model - which has sufficient flexibility to make it more viable as a reference monitor in commercial systems. The aim of this short paper is to introduce the result of [8] and its applications in an accessible way. Therefore, the presentation will be informal - although we will indicate where the technical details can be obtained. Our main objective is to illustrate the utility of, and generate some interest in, our approach.

The remainder of this paper is organised as follows. Section 2 presents the pre-requisite mathematical material in an informal way through the use of illustrative examples. Section 3 describes our model for separation of duty policies. Section 4 describes the hierarchical secure systems model and considers a special case of this model - HSS RBAC. In conclusion we consider the possibilities for future research.

2 Posets

A *partially ordered set* is a pair $\langle X, \leq \rangle$ such that for all $x, y, z \in X$,

$$\begin{aligned} x &\leq x; \\ x &\leq y \text{ and } y \leq x \text{ implies } x = y; \\ x &\leq y \text{ and } y \leq z \text{ implies } x \leq z. \end{aligned}$$

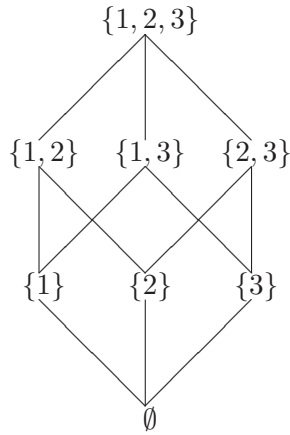
These properties are called *reflexivity*, *anti-symmetry* and *transitivity*, respectively.

X is a *chain* or *total order* if, for all $x, y \in X$, either $x \leq y$ or $x \geq y$. X is an *antichain* if, for all $x, y \in X$, $x \leq y$ only if $x = y$.

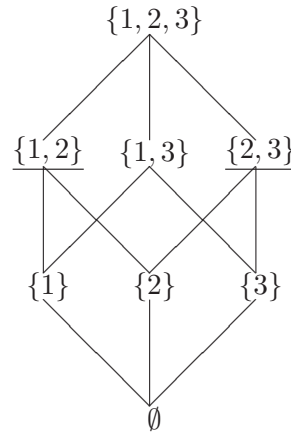
Given a poset X , a non-empty subset Y of X is an (*order*) *ideal* if for all $x \in X, y \in Y$, $x \leq y$ implies $x \in Y$. A non-empty subset Y of X is called an (*order*) *filter* if for all $x \in X, y \in Y$, $x \geq y$ implies $x \in Y$. Every filter, F , can be uniquely identified with an antichain, namely the set of *minimal elements* in the filter, which we will denote \underline{F} . Similarly, every ideal, I , can be uniquely identified with the antichain consisting of the set of *maximal elements* in the ideal, which we will denote \bar{I} . Figure 1 displays these characteristics of antichains, filters and ideals.

Posets are represented pictorially by Hasse diagrams. The nodes are the elements of the poset; if $x < y$ and there is no $z \in X$ such that $x < z < y$ then y is positioned above x and a line is drawn between the two nodes. Figure 1a shows a Hasse diagram of the power set of $\{1, 2, 3\}$. Figure 1b underlines the elements in the antichain $\{\{1, 2\}, \{2, 3\}\}$. Figures 1c and 1d underlines the elements of the corresponding ideal and filter, respectively.

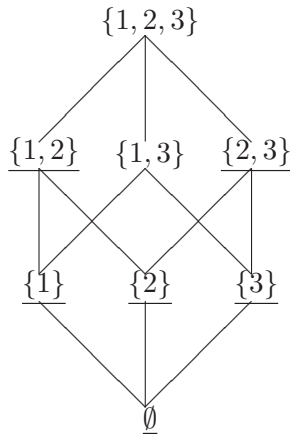
(a) The poset $\mathcal{P}(\{1, 2, 3\})$



(b) An antichain



(c) An ideal



(d) A filter

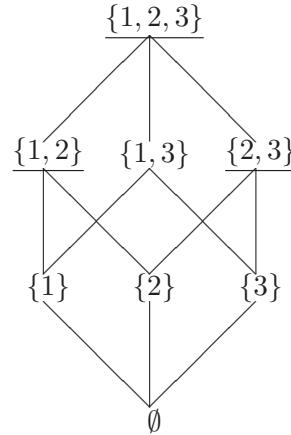


Figure 1: Hasse diagrams of a poset highlighting antichains, ideals and filters

Given a poset X we define $\mathcal{A}(X)$ to be the set of antichains in X , and for all $\alpha, \beta \in \mathcal{A}(X)$, we define [7]

$\alpha \preceq_1 \beta$ if, and only if, for all $a \in \alpha$, there exists $b \in \beta$ such that $a \leq b$ and

$\alpha \preceq_2 \beta$ if, and only if, for all $b \in \beta$, there exists $a \in \alpha$ such that $a \leq b$.

We next state the following result from [8].

Theorem 2.1 $\langle \mathcal{A}(X), \preceq_1 \rangle$ and $\langle \mathcal{A}(X), \preceq_2 \rangle$ are completions of X .

In other words $\mathcal{A}(X)$ is a complete lattice and the mapping $\phi : X \rightarrow \mathcal{A}(X)$, where $\phi(x) = \{x\}$, preserves the ordering of X in $\mathcal{A}(X)$. In practical terms this means that for every pair of elements $x, y \in X$ we can find a unique element $\alpha \in \mathcal{A}(X)$ such that, for $i = 1, 2$, $\phi(x) = \{x\} \preceq_i \alpha$ and $\phi(y) = \{y\} \preceq_i \alpha$.

Figure 2 shows an example of a poset X and the corresponding lattice $\mathcal{A}(X)$. (The nodes u , c , s and ts stand for **unclassified**, **classified**, **secret** and **top secret**, respectively.) We have omitted the comma delimiters in the sets because of space constraints. Note that, for example, $\{sC_1\}$, $\{s\}$ and $\{C_1\}$ are all antichains, and that $\{s\} \preceq_1 \{s, C_1\}$ and $\{C_1\} \preceq_1 \{s, C_1\}$.

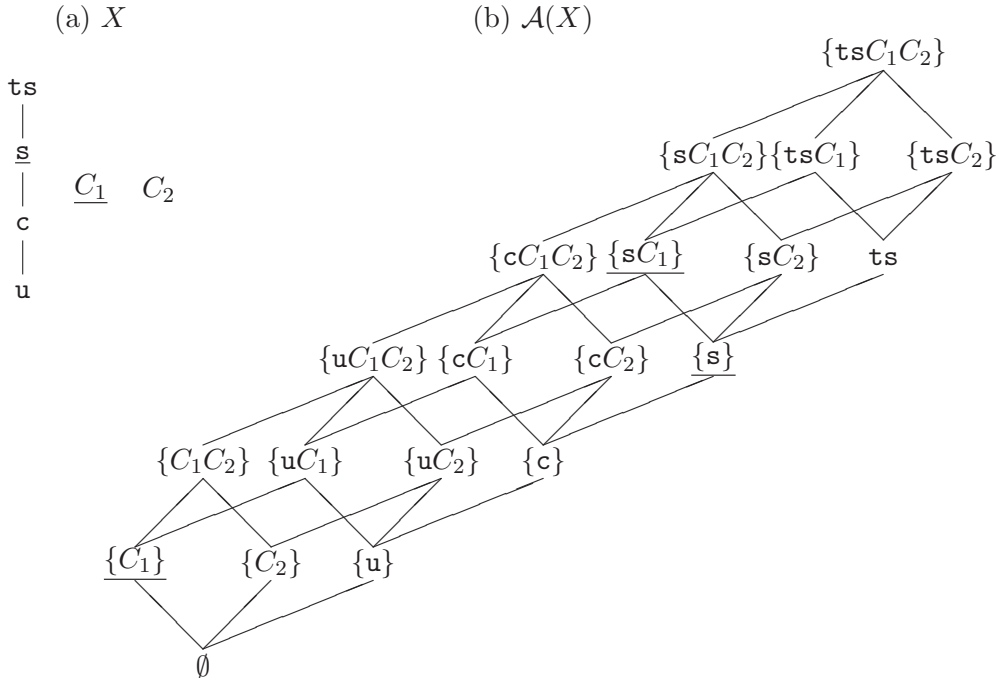


Figure 2: A “Bell-LaPadula poset” and its completion in the set of antichains

3 Separation of duty policies

Separation of duty policies have been of interest in access control modelling for several years. A review of the literature can be found in [1]. The purpose of a separation of duty policy is to specify what combination of access rights¹ is undesirable.

¹This is something of a simplification for the purposes of this paper. In general, we may want to consider other access control “artefacts”. For example, in a role-based access control model, it is necessary to specify undesirable combinations of roles.

For example, a Finance Department would partition duties in such a way that one person is responsible for producing payment files and another person is responsible for the cheque production. This can be modelled by saying that no subject (user) can execute both the payment files program and the cheque production program.

The most recent framework for considering separation of duty policies is the language RCL 2000 [1]. This language is used in a role-based access control model [18], although we believe it could easily be used with other models with appropriate modifications. The semantics of RCL 2000 are given by a translation to a restricted subset of first-order logic.

Our approach is more direct and uses a lattice of antichains with the ordering \preceq_2 [5]. (Our approach is also more general, as it can be used with more than one access control model.)

To illustrate our approach we will assume the reader is familiar with the protection matrix model [12]. The characteristic feature of this model is that the access rights a *subject*, s , has to an *object*, o , are given by the appropriate entry in the protection matrix. This entry is denoted $[s, o]$ and is a subset of R , where R is the set of access modes - read, write and execute, for example.

We will specify a separation of duty policy in terms of the set $M = \{m_1, m_2, \dots, m_n\}$ of *access right triples* which are derived from the protection matrix in the natural way, namely

$$(s, o, r) \in M \text{ if, and only if, } r \in [s, o].$$

We define a *conflict of interest policy* to be an antichain in the power set of M . (The reason why a conflict of interest policy should be an antichain is discussed in detail in [5].) An element of a conflict of interest policy is called a *conflict of interest constraint*. A conflict of interest policy, $\alpha = \{A_1, \dots, A_k\}$, where $A_i \subseteq S \times O \times R$, $1 \leq i \leq k$, is *violated* if $A_i \subseteq M$ for some i , $1 \leq i \leq k$. Intuitively, no conflict of interest constraint can be included in the protection environment, M .

Figure 3 shows the lattice of conflict of interest policies where $|M| = 3$. (For simplicity we identify m_i with the integer i . The power set of $\{1, 2, 3\}$ is shown in Figure 1.) That is, the set of conflict of interest policies can be viewed as the lattice $\mathcal{A}(\mathcal{P}(M))$. Note that we have a natural interpretation of the composition of two policies as the “meet” of two policies in the lattice. In other words the composition of α and β , the policy which implements both α and β , is the largest policy (with respect to \preceq_2) which is less than both α and β . (Formally, this is the *greatest lower bound* of α and β .) For example, the composition of the policies $\{\{2\}\}$ and $\{\{1, 2\}, \{1, 3\}, \{2, 3\}\}$ is $\{\{2\}, \{1, 3\}\}$ as shown in Figure 3.

Furthermore, we have a natural interpretation of the *strength* of a conflict of interest policy. Namely, if $\alpha \preceq_2 \beta$ then α is more *restrictive* or *stronger* than β .

We use the terminology “conflict of interest” because our definition can model policies which are usually regarded as being outside the scope of separation of duty policies. For example, the policy

$$\alpha = \{\{m_1\}, \dots, \{m_k\}\}$$

is violated if any of the triples m_1, \dots, m_k belong to M . In particular, $\alpha = \{\{(s, o, x)\}\}$ prohibits x from ever entering $[s, o]$ and thus prevents subject s executing (using the UNIX interpretation of the access right x) the file o . In other words, “conflict of interest” is preferred to separation of duty because we can model situations which are better described as conflicting with security requirements.

Most approaches to separation of duty policies assume it is sufficient to consider mutually exclusive pairs [1, 11]. In many cases this is certainly true, but there is no good reason to restrict one’s attention to this special case. Indeed, so-called *operational separation of duty policies* in role-based access control [1] can be regarded as sets of undesirable combinations of *capabilities*². There seems no reason to assume that such sets necessarily consist of only two elements. However, there are implementation considerations, which we consider in some detail in [5], which suggest it would be rather expensive computationally to use conflict of interest constraints of arbitrary size. We note that our ordering on policies means we can represent an arbitrary policy, α , as a policy, α' , in which every constraint is a pair, and such that α' is at least as strong a policy as α .

4 The hierarchical secure systems model

This model, which is based on the Bell-LaPadula model [2], was motivated by the following observations.

- Multi-level secure systems are usually too inflexible to be used in commercial environments [15].
- Commercial systems are usually based on discretionary access control models and hence it is difficult, if not impossible, to reason about the security implications of changes to the reference monitor. (The work in [9, 10, 15] is an attempt to address these issues.)
- The lattice of antichains using the ordering \preceq_1 derived from the disjoint union of the set of security labels, K , and the set of needs-to-know categories, C , is precisely the security lattice in the Bell-LaPadula model. This is shown explicitly in Figure 2 where there are only two needs-to-know categories, C_1 and C_2 .

By substituting the set of security labels for a set of *positions* representing an organisation hierarchy, and by using a *seniority function* which associates every subject and object with a level of seniority within the organisation, we obtain a model which exhibits the strong security properties of the Bell-LaPadula model with additional flexibility; the latter feature is provided by the more general framework of the position hierarchy.

²A capability is usually modelled as a row in the protection matrix [12]. We are treating a capability as an object-access right pair. This approach is similar to that adopted by role-based access control models [1].

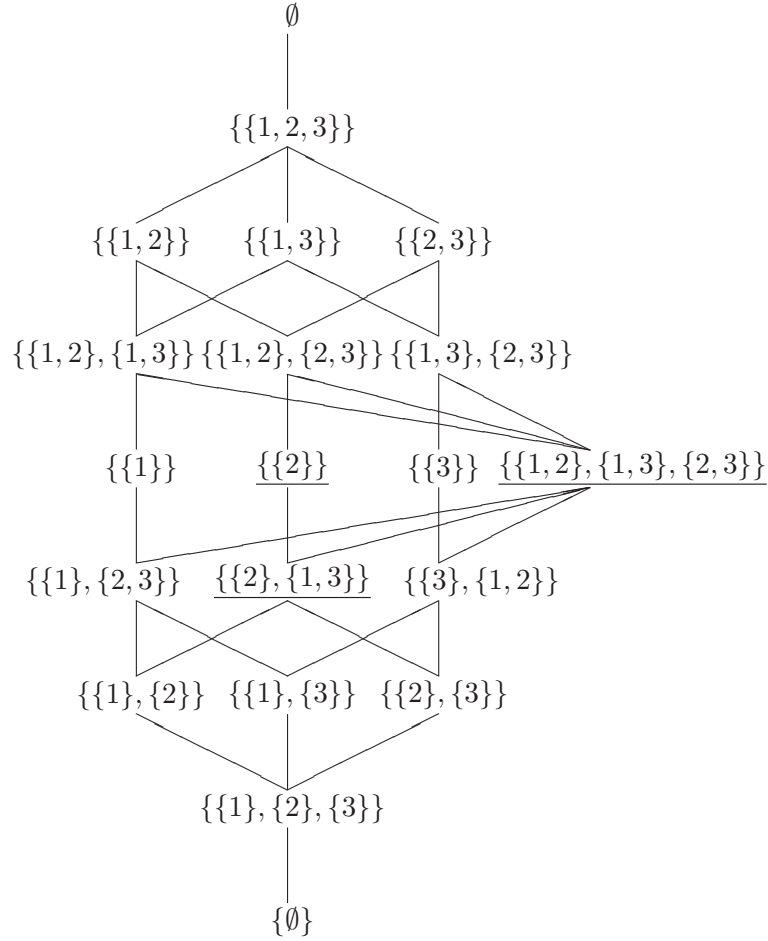


Figure 3: The lattice $\langle \mathcal{A}(\mathcal{P}(M)), \preceq_2 \rangle$ where $M = \{1, 2, 3\}$

The resulting model is called the hierarchical secure systems model. The basic framework is presented in [6]. Therein we consider special cases of the model, of which the Bell-LaPadula model is the most obvious.

The hierarchical secure systems model is too general to be of immediate use; from a practical point of view we need to consider systems with particular access modes and particular seniority policies. (Indeed we are in a similar situation to that of Bell and LaPadula after the completion of the first volume of their seminal paper [2]. In the second volume they introduce specific access modes, the simple security property and the *-property [3].) The first step in this direction is the HSS RBAC model, a role-based access control model with stronger security properties than those in the literature. (HSS stands for hierarchical secure systems.)

HSS RBAC has the usual features of a role-based access control model. In particular, it makes use of a role hierarchy, a user-role assignment relation and a permission-role assignment relation. The main additional features of HSS RBAC are:

- A position hierarchy
- A seniority function, ϕ , which associates each role, user and permission with a position in the hierarchy
- A policy monitor:
 - if a user u wants to activate the role r then the role activation property, $\phi(r) \preceq_1 \phi(u)$, must be satisfied;
 - if a role r wishes to assign a user u to another role r' then the user assignment property, $\phi(r') \preceq_1 \phi(u) \prec_1 \phi(r)$, must be satisfied.
- A reference monitor based on a role-based access control model which includes
 - a role hierarchy
 - a user-role assignment relation
 - a permission-role assignment relation

If the requirements of the policy monitor are satisfied, the reference monitor checks that the relevant assignments are available. This is analogous to the Bell-LaPadula model in which a request by s to access o in mode r is only granted if the requirements of the information flow policy are satisfied and $r \in [s, o]$.

In [6] we show that HSS RBAC displays the following advantages over existing role-based access control models [6].

- More natural inheritance of permissions and more natural role ordering
- More flexibility and greater control over the assignment of users to roles
- Ease of administration and reduced complexity of decision procedures
- Stronger security properties - for example, it is always possible to answer the question “can a given user be assigned to a given role”
- Stronger correlation of access control model with organisation characteristics

The disadvantage of our approach compared with conventional role-based access control models is a certain loss of flexibility in the assignment of users to roles. It is also possible that the maintenance of an additional hierarchy and relations (namely, the position hierarchy, user-position relation and role-position relation) imposes an unsatisfactory overhead. However, we would argue that the most well-known administration model for role-based access control, ARBAC97 [17], needs to introduce at least as much “machinery”, the implications of which are considerably less easy to understand than those of our model.

5 Conclusion

Our immediate concern is to extend the work on HSS RBAC in a style similar to that of the Bell-LaPadula model. Namely, to provide a security theorem and a set of primitive operations (rules) which preserve secure states of the system [3].

We then intend to investigate other special cases of the model. We are particularly interested in determining whether we can treat the typed access matrix model [16] as a special case of the hierarchical secure systems model.

Acknowledgements The work of Jason Crampton is supported by EPSRC Award 98317878.

References

- [1] AHN, G.-J., AND SANDHU, R. Role-based authorization constraints specification. *ACM Transactions on Information and System Security* 3, 4 (2000).
- [2] BELL, D., AND LAPADULA, L. Secure computer systems: Mathematical foundations. Tech. Rep. MTR-2547, Volume I, Mitre Corporation, 1973.
- [3] BELL, D., AND LAPADULA, L. Secure computer systems: A mathematical model. Tech. Rep. MTR-2547, Volume II, Mitre Corporation, 1973.
- [4] BRUALDI, R. *Introductory Combinatorics*. Prentice Hall, New Jersey, 1999.
- [5] CRAMPTON, J., AND LOIZOU, G. The structural complexity of conflict of interest policies. In preparation.
- [6] CRAMPTON, J., AND LOIZOU, G. Hierarchical secure systems: A preliminary description. Tech. Rep. BBKCS-00-18, Birkbeck College, University of London, 2000.
- [7] CRAMPTON, J., AND LOIZOU, G. Two partial orders on the set of antichains. Tech. Rep. BBKCS-00-05, Birkbeck College, University of London, 2000.
- [8] CRAMPTON, J., AND LOIZOU, G. The completion of a poset in a lattice of antichains. *International Mathematical Journal* (2001). Accepted for publication.
- [9] CRAMPTON, J., LOIZOU, G., AND O'SHEA, G. Evaluating access control. *Submitted* (1999).
- [10] CRAMPTON, J., LOIZOU, G., AND O'SHEA, G. A logic of access control. *The Computer Journal* 44, 1 (2001). To appear.

- [11] GAVRILA, S., AND BARKLEY, J. 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* (Fairfax, Virginia, 1998), pp. 81–90.
- [12] HARRISON, M., RUZZO, W., AND ULLMAN, J. Protection in operating systems. *Communications of the ACM* 19, 8 (1976), 461–471.
- [13] LAMPSON, B. Protection. *ACM Operating Systems Review* 8 (1974), 437–443.
- [14] LLOYD, J. *Foundations of Logic Programming*. Springer-Verlag, London, 1984.
- [15] O’SHEA, G. *Access Control in Operating Systems*. PhD thesis, Birkbeck College, University of London, 1997.
- [16] SANDHU, R. The typed access matrix model. In *Proceedings of IEEE Symposium on Security and Privacy* (1992), IEEE, pp. 122–136.
- [17] SANDHU, R., BHAMIDIPATI, V., AND MUNAWER, Q. The ARBAC97 model for role-based administration of roles. *ACM Transactions on Information and System Security* 1, 2 (1999), 105–135.
- [18] SANDHU, R., COYNE, E., FEINSTEIN, H., AND YOUMAN, C. Role-based access control models. *IEEE Computer* 29, 2 (1996), 38–47.
- [19] SANDHU, R., FERRAILOLO, D., AND KUHN, D. The NIST model for role-based access control: Towards a unified standard. In *Proceedings of the 5th ACM Workshop on Role-Based Access Control* (Phoenix, Arizona, USA, 2000). <http://www.acm.org/sigsac/nist.pdf>.
- [20] SPERNER, E. Ein Satz über Untermengen einer endlichen Menge. *Mathematische Zeitschrift* 27 (1928), 544–548.