

# A formal comparison of the Bell & LaPadula and RBAC models

Lionel Habib, Mathieu Jaume  
SPI LIP6 University Paris 6  
104 av du Président Kennedy, 75016, Paris, France  
Lionel.Habib@lip6.fr Mathieu.Jaume@lip6.fr

Charles Morisset  
UNU-IIST  
P.O. Box 3058, Macau SAR, China  
morisset@iist.unu.edu

## Abstract

*In this paper we address the problem of comparing access control models. Indeed, many access control models can be found in the literature and in order to choose one model for a particular context, some tools helping such a choice are needed. We develop here a complete example allowing to compare (in a formal way) the Bell and LaPadula (BLP) model and the Role-Based (RBAC) model. In order to achieve this goal, we first express these models in a uniform way, then we introduce concepts (mostly based on simulations) allowing to compare access control models.*

## 1. Introduction

Access control is any mechanism by which a system grants or revokes rights to or from active entities, the subjects, to access some passive entities, the objects, or to perform some action. In [14], Lampson introduces the basic concepts: active and passive entities, access matrix, etc. They serve in the definition of many access control models such as [9, 2, 4, 6]. Such policies and models describe their own notion of information system and the accesses they grant. In fact, such approaches cannot be easily reused when considering new models or even variants of these models. Indeed, although many access control policies can now be found in the literature, their descriptions often suffer from a lack of precision: formal and mathematical specifications are rare. Furthermore, these policies are not described within a common framework and it is rather difficult to extract from these developments some methodological guidelines. In this paper, we first show how to specify and to define an access control model regardless of any specific context, by defining two classical models in a uniform way. We consider here the Bell & LaPadula model (BLP) [15] and the Role-Based Access Control model (RBAC) [6] in its RBAC96 variation. Then, we introduce a general way to compare two access control models and we illustrate our method by considering the BLP and RBAC models. Our

approach, based on the notion of simulation of implementations, provides some formal tools to express the reusing of implementations and some abstract notions to help the comparison of models. In [18, 8], we have defined and compared several classical models by following our approach. There exist some papers focusing on comparisons and translations between policies [22, 1, 19], but here again, these developments are not expressed within a common framework. Hence, it is rather difficult to compose or to compare these translations. Our approach provides a common framework allowing to express the comparison of two arbitrary access control models.

This framework is the result of several projects [7, 10, 11], done during the last few years and whose main objective is to develop a formal library of access control policies within the Focal [21] integrated development environment. However, such a development is rather technical and time-consuming. Hence, in order to ease formal developments of access control models, the main conclusion of this work leads us to define and to implement an abstract framework which is expressive enough to define access control models as instances of this framework, and to reason about them in a mathematical setting. A preliminary and different version of this framework has been defined in [12]. In this paper, we use a new version of this framework, based on the definitions given in [18]. It is based on more intuitive concepts thus easing the specification and implementation of access control policies. Furthermore, it generalizes our framework and extend our results. Due to space limitations, some technical material of our work is omitted here but it can be found in [13, 18, 8]. Furthermore, the aim of this paper is not to present the whole theory on which our framework is based, but rather to illustrate it over a concrete example.

## 2. Defining access control models

In this section, we present a way to define access control systems, based on two main concepts: policy and model. The policy is the description of the system on which it is enforced, defined as a state machine together with the notion

of secure states. Hence, an access control policy is considered here as a functional property that a state machine must satisfy. Of course, the definition of the policy also includes all the information relevant to the definition of the system, such as subjects, objects, security information, etc. At this point, a policy can be seen as “static”, since it is expressed over states, and states are “snapshot” of the system. We then introduce the notion of model, which is basically a policy together with a set of requests (and, as we will see later, the semantics of these requests). Last, it is possible to define an implementation (or several) for a model, as a transition function and a set of initial states. Intuitively, this implementation corresponds to a reference monitor and should be proved correct with respect to the security policy, that is returning a secure state for any secure state and any request.

**Entities** We first define the main entities of a system:  $\mathcal{S}$  is the set of subjects (active entities initiating actions in the system),  $\mathcal{O}$  is the set of objects (passive entities on which actions are made) and  $\mathcal{A}$  is the set of access modes (read, write, append, etc.). In this paper, we represent an access by a triple  $(s, o, a)$  expressing that a subject  $s$  accesses an object  $o$  according to the access mode  $a$ . Hence, we define the set of accesses  $\mathbb{A}$  as the cartesian product  $\mathcal{S} \times \mathcal{O} \times \mathcal{A}$ . Other approaches are possible to represent accesses. For example, in order to deal with “joint access” of a group of subjects over an object, as it is done in [17],  $\mathbb{A}$  can be defined as  $(\wp(\mathcal{S}) \setminus \{\emptyset\}) \times \mathcal{O} \times \wp(\mathcal{A})$ .

**BLP policy** The BLP model [15, 2] constrains accesses by considering levels of security associated with subjects and objects. We write  $\rho_{\text{blp}} = (\mathcal{L}, \preceq, \sqcup, \sqcap)$  for the *lattice of levels of security*. In [15, 2],  $\rho_{\text{blp}}$  is defined as the product lattice of a lattice of classification and a powerset lattice of needs-to-know, but we do not need here such details. Each subject and object is associated with a security level specified by the *security functions*  $f_s : \mathcal{S} \rightarrow \mathcal{L}$  and  $f_o : \mathcal{O} \rightarrow \mathcal{L}$ . The lattice of levels of security is supposed to be constant during the life of the system and is called a *security parameter*, while security levels of subjects and objects may vary. This distinction between security parameters and security functions is not a restriction, but a way to specify what can be modified and what is supposed to be constant. In this context, a state  $\sigma \in \Sigma_{\text{blp}}$  is a tuple  $\sigma = (m, f_s, f_o)$  where  $m \subseteq \mathbb{A}$  is the set of current accesses done in the system. We do not consider here the discretionary part of the BLP model and the set  $\mathcal{A}$  only contains the access modes  $r$  (for read) and  $w$  (for write). This policy is specified by a predicate  $\Omega_{\text{blp}}$  over states as follows. Given a state  $\sigma = (m, f_s, f_o)$ ,  $\Omega_{\text{blp}}(\sigma)$  holds iff the two following properties are satisfied.

$$\begin{aligned} \forall s \in \mathcal{S} \forall o \in \mathcal{O} \quad (s, o, r) \in m &\Rightarrow f_o(o) \preceq f_s(s) \\ \forall s \in \mathcal{S} \forall o_1, o_2 \in \mathcal{O} \\ (s, o_1, r) \in m \wedge (s, o_2, w) \in m &\Rightarrow f_o(o_1) \preceq f_o(o_2) \end{aligned}$$

The second property prevents the copy of an object to a lower security level by a malicious subject. We write  $\mathbb{P}_{\text{blp}}[\rho_{\text{blp}}] = (\mathcal{S}, \mathcal{O}, \mathcal{A}, \Sigma_{\text{blp}}, \Omega_{\text{blp}})$  for the BLP policy.

**RBAC policy** Role-Based Access Control models are a set of fairly new models first introduced in the ninety’s. The RBAC92 model [6] introduces the concept of roles, and RBAC96 [23] refines RBAC92 thanks to the addition of the users notion (different from the subjects one) and a roles hierarchy defined as a partial order. Each user of the system is associated with roles, themselves associated with permissions. An access is granted if the user requesting it has activated a role associated with the permission corresponding to this access. We write  $\rho_{\text{rbac}} = (\mathbf{U}, \mathbf{R}, \leq_{\mathbf{R}})$  for the security parameter of RBAC, where  $\mathbf{U}$  is the set of users and  $\leq_{\mathbf{R}}$  is the partial order over the set of roles  $\mathbf{R}$ . A state  $\sigma \in \Sigma_{\text{rbac}}$  is a tuple  $\sigma = (m, \text{user}, \mathbf{UA}, \mathbf{PA}, \text{roles})$  where  $m$  is the set of current accesses,  $\text{user} : \mathcal{S} \rightarrow \mathbf{U}$  allows to know the user corresponding to a subject,  $\mathbf{UA} \subseteq \mathbf{U} \times \mathbf{R}$  is the relation specifying which users can activate which roles (and the roles lower to them according to  $\leq_{\mathbf{R}}$ ),  $\mathbf{PA} \subseteq (\mathcal{O} \times \mathcal{A}) \times \mathbf{R}$  is the relation associating permissions (i.e. pairs  $(o, a) \in \mathcal{O} \times \mathcal{A}$ ) to roles, and  $\text{roles} : \mathcal{S} \rightarrow \wp(\mathbf{R})$  specifies the set of roles that have been activated by a subject. In this formalization,  $\text{user}$ ,  $\mathbf{UA}$ ,  $\mathbf{PA}$  and  $\text{roles}$  are the security functions. The RBAC policy is specified by the predicate  $\Omega_{\text{rbac}}$  as follows. Given a state  $\sigma = (m, \text{user}, \mathbf{UA}, \mathbf{PA}, \text{roles})$ ,  $\Omega_{\text{rbac}}(\sigma)$  holds iff the two following properties are satisfied.

$$\begin{aligned} \forall s \in \mathcal{S} \quad \text{roles}(s) &\subseteq \text{ER}(s, \mathbf{UA}) \\ \forall s \in \mathcal{S} \quad \forall o \in \mathcal{O} \quad \forall a \in \mathcal{A} \\ (s, o, a) \in m &\Rightarrow (o, a) \in \text{EP}(s, \mathbf{PA}, \text{roles}) \end{aligned}$$

where, given a subject  $s \in \mathcal{S}$ ,  $\text{ER}(s, \mathbf{UA})$  is the set of roles that  $s$  can activate according to  $\mathbf{UA}$  and  $\text{EP}(s, \mathbf{PA}, \text{roles})$  is the set of permissions associated with the roles activated by  $s$ :

$$\begin{aligned} \text{ER}(s, \mathbf{UA}) &= \left\{ r \in \mathbf{R} \mid \begin{array}{l} \exists r' \in \mathbf{R} \quad r \leq_{\mathbf{R}} r' \\ \wedge (\text{user}(s), r') \in \mathbf{UA} \end{array} \right\} \\ \text{EP}(s, \mathbf{PA}, \text{roles}) &= \\ \bigcup_{r \in \text{roles}(s)} &\left\{ (o, a) \in (\mathcal{O} \times \mathcal{A}) \mid \begin{array}{l} \exists r'' \in \mathbf{R} \quad r'' \leq_{\mathbf{R}} r \\ \wedge ((o, a), r'') \in \mathbf{PA} \end{array} \right\} \end{aligned}$$

We write  $\mathbb{P}_{\text{rbac}}[\rho_{\text{rbac}}] = (\mathcal{S}, \mathcal{O}, \mathcal{A}, \Sigma_{\text{rbac}}, \Omega_{\text{rbac}})$  for the RBAC policy.

**Requests** As we said previously, a language of requests provides to the users of a system a way to access to objects. We write  $\mathcal{R}$  for the set of requests. Most access control models consider at least the set  $\mathcal{R}^{\text{acc}} = \{(+, s, o, a), (-, s, o, a)\}$  allowing to express that the subject  $s$  asks to get (+) or to release (-) an access over the

object  $o$  according to the access mode  $a$ . Depending of the access control model, there can also exist some “administrative” requests allowing to modify security functions of a state (for example, requests allowing to change the security levels of objects in the case of the BLP model, or to change the active roles of a subject in the case of the RBAC model). In this paper, we only consider requests in  $\mathcal{R}^{acc}$ . As discussed in the conclusion, when comparing models, taking into account administrative requests requires some more complicated formal tools (indeed, this leads to compare two different languages of requests).

We make here a clear distinction between accesses and requests. An access is the internal representation of actions currently done in the system and is authorized or not according to the security policy. A request is an action that a user has to submit and is granted or not by an implementation. However, requests are usually strongly related to accesses, and to explicit this relation, we introduce a notion of “weak” semantics of requests as a relation  $\llbracket \mathcal{R} \rrbracket_{\Sigma} \subseteq \mathcal{R} \times \Sigma$ . Given a request  $R$  and a state  $\sigma$ , the statement  $(R, \sigma) \in \llbracket \mathcal{R} \rrbracket_{\Sigma}$  characterizes the properties that a state  $\sigma$  must satisfy when it is obtained by applying (in a successful way) the request  $R$  over another state. For  $\mathcal{R}^{acc}$ , we can define  $\llbracket \mathcal{R}^{acc} \rrbracket_{\Sigma}$  as follows:

$$\begin{aligned} (\langle +, s, o, a \rangle, \sigma) \in \llbracket \mathcal{R}^{acc} \rrbracket_{\Sigma} &\Leftrightarrow (s, o, a) \in \Lambda(\sigma) \\ (\langle -, s, o, a \rangle, \sigma) \in \llbracket \mathcal{R}^{acc} \rrbracket_{\Sigma} &\Leftrightarrow (s, o, a) \notin \Lambda(\sigma) \end{aligned}$$

where  $\Lambda(\sigma)$  denotes the set of all current accesses in  $\sigma$ . Note that such an approach to express a part of the semantics of requests only specifies the properties that a state must satisfy but does not describe how such a state has been changed. We introduce in [13, 18] a semantical characterisation of such modifications. Due to space limitation, we omit here this technical part which is not essential to understand our comparison mechanism.

### Access control models and their implementations

Given a security parameter  $\rho$ , an access control model  $\mathbb{M}[\rho]$  is defined by a tuple  $\mathbb{M}[\rho] = (\mathbb{P}[\rho], \llbracket \mathcal{R} \rrbracket_{\Sigma})$  where  $\mathbb{P}[\rho] = (\mathcal{S}, \mathcal{O}, \mathcal{A}, \Sigma, \Omega)$  is an access control policy,  $\mathcal{R}$  is a set of requests, and  $\llbracket \mathcal{R} \rrbracket_{\Sigma} \subseteq \mathcal{R} \times \Sigma$  is a relation specifying the semantics of requests. For example, we write  $\mathbb{M}_{blp}[\rho_{blp}] = (\mathbb{P}_{blp}[\rho_{blp}], \llbracket \mathcal{R}^{acc} \rrbracket_{\Sigma_{blp}})$  (resp.  $\mathbb{M}_{rbac}[\rho_{rbac}] = (\mathbb{P}_{rbac}[\rho_{rbac}], \llbracket \mathcal{R}^{acc} \rrbracket_{\Sigma_{rbac}})$ ) for the BLP model (resp. for the RBAC model).

Implementing a model  $\mathbb{M}[\rho]$  consists in defining both a set  $\Sigma_I$  of initial states and a transition function  $\tau : \mathcal{R} \times \Sigma \rightarrow \mathcal{D} \times \Sigma$  (where  $\mathcal{D} = \{\text{yes}, \text{no}\}$  are the answers) which allows to move from a state to another state of the system according to a request in  $\mathcal{R}$ . We write  $(\tau, \Sigma_I)$  such an implementation and  $\Gamma_{\tau}(E)$  the set of reachable states by  $\tau$  from states occurring in  $E$ . For example, given the set of initial states  $\Sigma_I^{blp} = \{\sigma \in \Sigma_{blp} \mid \Lambda(\sigma) = \emptyset\}$ , we introduce

the implementation  $(\tau_{blp}, \Sigma_I^{blp})$  of  $\mathbb{M}_{blp}[\rho_{blp}]$  where  $\tau_{blp}$  is defined in table 1. Similarly, given the set of initial states  $\Sigma_I^{rbac} = \{\sigma \in \Sigma_{rbac} \mid \Lambda(\sigma) = \emptyset\}$ , we introduce the implementation  $(\tau_{rbac}, \Sigma_I^{rbac})$  of  $\mathbb{M}_{rbac}[\rho_{rbac}]$  where  $\tau_{rbac}$  is defined in table 2.

In [18, 8], these implementations are proved correct according to both the policy and the semantics of requests. More formally, for each of them, we prove that each state reachable from an initial state is secure (i.e.  $\Gamma_{\tau}(\Sigma_I) \subseteq \{\sigma \in \Sigma \mid \Omega(\sigma)\}$ ) and that for all  $\sigma_1, \sigma_2 \in \Sigma$ , and  $R \in \mathcal{R}$ , if  $\tau(R, \sigma_1) = (\text{yes}, \sigma_2)$ , then  $(R, \sigma_2) \in \llbracket \mathcal{R} \rrbracket_{\Sigma}$ .

$$\begin{array}{l} \tau_{blp}(R, (m, f_s, f_o)) \\ \\ = \left\{ \begin{array}{l} (\text{yes}, (m \cup \{(s, o, r)\}, f_s, f_o)) \\ \text{if } R = \langle +, s, o, r \rangle \\ \wedge f_o(o) \preceq f_s(s) \\ \wedge \left\{ \begin{array}{l} o' \in \mathcal{O} \mid (s, o', w) \in m \wedge \\ \neg(f_o(o) \preceq f_o(o')) \end{array} \right\} = \emptyset \\ \\ (\text{yes}, (m \cup \{(s, o, w)\}, f_s, f_o)) \\ \text{if } R = \langle +, s, o, w \rangle \\ \wedge \left\{ \begin{array}{l} o' \in \mathcal{O} \mid (s, o', r) \in m \wedge \\ \neg(f_o(o') \preceq f_o(o)) \end{array} \right\} = \emptyset \\ \\ (\text{yes}, (m \setminus \{(s, o, a)\}, f_s, f_o)) \\ \text{if } R = \langle -, s, o, a \rangle \\ \\ (\text{no}, (m, f_s, f_o)) \text{ otherwise} \end{array} \right. \end{array}$$

**Table 1. Implementation of the BLP policy**

$$\begin{array}{l} \tau_{rbac}(R, (m, user, UA, PA, roles)) \\ \\ = \left\{ \begin{array}{l} (\text{yes}, (m \cup \{(s, o, a)\}, user, UA, PA, roles)) \\ \text{if } R = \langle +, s, o, a \rangle \\ \wedge (o, a) \in EP(s, PA, roles) \\ \\ (\text{yes}, (m \setminus \{(s, o, a)\}, user, UA, PA, roles)) \\ \text{if } R = \langle -, s, o, a \rangle \\ \\ (\text{no}, (m, user, UA, PA, roles)) \text{ otherwise} \end{array} \right. \end{array}$$

**Table 2. Implementation of the RBAC96 Model**

### 3. Comparison of the two models

In this section, we introduce a preorder over access control models. Roughly speaking, an access control model  $\mathbb{M}_1[\rho_1]$  is said to be more restrictive than an access control model  $\mathbb{M}_2[\rho_2]$  iff for any correct implementation of  $\mathbb{M}_1[\rho_1]$ , we can define a correct implementation of  $\mathbb{M}_2[\rho_2]$  such that the latter simulates the former. However, by following this approach, comparing two access control models requires to consider all the implementations of an access control model. Nevertheless, since in practice the simulation relations used to compare access control models seem to satisfy some (“good”) supplementary properties, in [13, 18], we prove results allowing to make this comparison without considering all the implementations when the simulation relation used satisfies such properties.

Our approach is based on the classical notion of simulations. Given two transition functions  $\tau_1 : \mathcal{R} \times \Sigma_1 \rightarrow \mathcal{D} \times \Sigma_1$  and  $\tau_2 : \mathcal{R} \times \Sigma_2 \rightarrow \mathcal{D} \times \Sigma_2$ ,  $\tau_2$  simulates  $\tau_1$ , and we write  $\tau_1 \stackrel{\kappa_\Sigma}{\sim} \tau_2$ , iff there exists a relation  $\kappa_\Sigma \subseteq \Sigma_1 \times \Sigma_2$  such that:

$$\begin{aligned} & \forall \sigma_1, \sigma'_1 \in \Sigma_1 \quad \forall \sigma_2 \in \Sigma_2 \quad \forall R \in \mathcal{R} \quad \forall d \in \mathcal{D} \\ & ((\sigma_1, \sigma_2) \in \kappa_\Sigma \wedge \tau_1(R, \sigma_1) = (d, \sigma'_1)) \\ & \Rightarrow (\exists \sigma'_2 \in \Sigma_2 \quad (\sigma'_1, \sigma'_2) \in \kappa_\Sigma \wedge \tau_2(R, \sigma_2) = (d, \sigma'_2)) \end{aligned}$$

We extend this definition to implementations of access control models:  $(\tau_2, \Sigma_2^1)$  simulates  $(\tau_1, \Sigma_1^1)$ , and we write  $(\tau_1, \Sigma_1^1) \stackrel{\kappa_\Sigma}{\sim} (\tau_2, \Sigma_2^1)$ , iff there exists a relation  $\kappa_\Sigma \subseteq \Sigma_1 \times \Sigma_2$  such that:

$$\tau_1 \stackrel{\kappa_\Sigma}{\sim} \tau_2 \wedge \forall \sigma_1 \in \Sigma_1^1 \quad \exists \sigma_2 \in \Sigma_2^1 \quad (\sigma_1, \sigma_2) \in \kappa_\Sigma$$

We are now in position to express in a formal way that an access control model  $\mathbb{M}_1[\rho_1]$  is more restrictive than a model  $\mathbb{M}_2[\rho_2]$  iff each correct implementation of  $\mathbb{M}_1[\rho_1]$  can be simulated by a correct implementation of  $\mathbb{M}_2[\rho_2]$ . However, the simulation relation used to simulate implementations has to satisfy some supplementary properties. Indeed, for example, considering the relation  $\kappa_\Sigma = \Sigma_1 \times \Sigma_2$  as a simulation relation could lead to prove that any model is more restrictive than any other model. Due to space limitation, we do not detail here such technical properties over simulation relations.

Comparing two models mostly consists in defining a correspondence between formalisms of these models. This kind of interpretation is presented in a more or less formal way in the literature. For instance, [20] presents intuitively how RBAC can be used to implement the Multics interpretation of the BLP model. We sketch here a different and fully formalized approach. First, we show how to obtain the security parameter  $\rho_{\text{rbac}}$  for RBAC from the set of subjects and the lattice of levels of security  $\rho_{\text{blp}} = (\mathcal{L}, \preceq, \sqcup, \sqcap)$  of the BLP model: at each security level  $l$  of  $\mathcal{L}$  matches a role that has the same name, and the partial order over the set of

roles is identical to the one of the lattice. Hence, we have  $\rho_{\text{rbac}} = (\mathcal{S}, \mathcal{L}, \preceq)$ . Now, we have to show how to “translate” security functions of the BLP model into security functions of the RBAC model.

– The function *user* associates a user to a subject in the RBAC model. Its translation is the identity function (i.e.  $\forall s \in \mathcal{S} \quad \text{user}(s) = s$ ).

– The relation **UA** is defined as the set  $\{(s, f_s(s)) \mid s \in \mathcal{S}\}$ . In this set, one unique role is connected to each subject as only one security level is attributed to each subject.

– The relation **PA** is defined as the set  $\{((o, r), f_o(o)), ((o, \mathbf{w}), f_o(o)), ((o, \mathbf{w}), \perp) \mid o \in \mathcal{O}\}$ . Here, the main difficulty comes from the write access. Indeed, within the BLP policy, if a subject does not read any object, he is allowed to write into any object. Hence, in this case, within the RBAC policy, any role should have the write permission over any object. Thanks to the partial order  $\preceq$  over the roles, it suffices to add the write permission to the role  $\perp$ . Of course, when a subject has a read access over an object, the write access is constrained according to the BLP policy as it is expressed in the second clause of the predicate  $\Omega_{\text{rbac}}$  defined below.

– For all subject  $s$ ,  $\text{roles}(s) = \{f_s(s)\}$  which means that a subject always activate the role corresponding to his security level.

These definitions lead to define the relation  $\kappa_\Sigma \subseteq \Sigma_{\text{blp}} \times \Sigma_{\text{rbac}}$  as follows:

$$\begin{aligned} & \forall \sigma_1 = (m_1, f_s, f_o) \in \Sigma_{\text{blp}} \\ & \forall \sigma_2 = (m_2, \text{user}, \mathbf{UA}, \mathbf{PA}, \text{roles}) \in \Sigma_{\text{rbac}} \\ & (\sigma_1, \sigma_2) \in \kappa_\Sigma \Leftrightarrow \\ & \left( \begin{array}{l} m_1 = m_2 \\ \wedge \forall s \in \mathcal{S} \quad f_s(s) = l \Leftrightarrow \\ \quad \mathbf{UA} = \{(s, l)\} \wedge \text{roles}(s) = \{l\} \\ \wedge \forall o \in \mathcal{O} \quad f_o(o) = l \Leftrightarrow \\ \quad \mathbf{PA} = \{((o, r), l), ((o, \mathbf{w}), l), ((o, \mathbf{w}), \perp)\} \end{array} \right) \end{aligned}$$

We can now introduce the security predicate  $\Omega_{\text{rbac}}$ , which is a translation of the predicate  $\Omega_{\text{blp}}$  in the RBAC formalism. Given a state  $\sigma = (m, \text{user}, \mathbf{UA}, \mathbf{PA}, \text{roles})$ , where its security functions are defined as previously,  $\Omega_{\text{rbac}}(\sigma)$  holds iff the two following properties are satisfied.

$$\forall s \in \mathcal{S} \quad \forall o \in \mathcal{O} \quad (s, o, r) \in m \Rightarrow \left( \forall r, r' \in \mathcal{L} \left( \begin{array}{l} r \in \text{roles}(s) \\ \wedge ((o, r), r') \in \mathbf{PA} \end{array} \right) \Rightarrow r' \preceq r \right)$$

$$\forall s \in \mathcal{S} \quad \forall o_1, o_2 \in \mathcal{O} \quad (s, o_1, r) \in m \wedge (s, o_2, \mathbf{w}) \in m \Rightarrow \left( \forall r, r' \in \mathcal{L} \left( \begin{array}{l} ((o_1, r), r) \in \mathbf{PA} \wedge \\ \left( r' = \bigsqcup \left\{ r'' \in \mathcal{L} \mid \right. \right. \right. \\ \left. \left. \left. ((o_2, \mathbf{w}), r'') \in \mathbf{PA} \right\} \right) \right) \Rightarrow r \preceq r' \end{array} \right)$$

The first property states that if a subject  $s$  reads an object  $o$  then the role  $r'$  associated with  $o$ 's read permission is below

the role  $r$  associated with  $s$ . The second property states that if a subject  $s$  reads an object  $o_1$  and writes into an object  $o_2$  then the role  $r$  associated with  $o_1$ 's read permission is below the supremum of the roles associated with  $o_2$ 's write permission (i.e. below the role  $r'$  corresponding to  $o_2$ 's security level given that the supremum is computed over a set containing at most two roles,  $r'$  and  $\perp$ ).

This definition of  $\Omega_{\text{rblp}}$  allows us to show that given a state  $\sigma \in \Sigma_{\text{rbac}}$ , if  $\Omega_{\text{rblp}}(\sigma)$  is satisfied then  $\Omega_{\text{rbac}}(\sigma)$  is satisfied. Moreover, we prove that the relation  $\kappa_{\Sigma}$  preserves the predicate  $\Omega_{\text{rblp}}$  and the relation  $\|\mathcal{R}^{acc}\|_{\Sigma}$ , i.e. given a state  $\sigma_1 \in \Sigma_{\text{blp}}$ , a state  $\sigma_2 \in \Sigma_{\text{rbac}}$  and a request  $R \in \mathcal{R}^{acc}$  if  $(\sigma_1, \sigma_2) \in \kappa_{\Sigma}$ ,  $(R, \sigma_1) \in \|\mathcal{R}^{acc}\|_{\Sigma_{\text{blp}}}$  and  $\Omega_{\text{rblp}}(\sigma_1)$  are verified then  $(R, \sigma_2) \in \|\mathcal{R}^{acc}\|_{\Sigma_{\text{rbac}}}$  and  $\Omega_{\text{rblp}}(\sigma_2)$  are satisfied. The formal proofs are given in [8]. Hence, a general theorem in [18] allows us to conclude from those results that the BLP model is more restrictive than the RBAC model. Furthermore, in order to show that the BLP model is strictly more restrictive than the RBAC model, we prove in [8] that the RBAC model is not more restrictive than the BLP model. The intuition is that there exists RBAC states that cannot be simulated by any BLP states. Indeed, we can define an RBAC state where a subject  $s$  has no rights over an object  $o$ , whereas in the BLP model a subject  $s$  has always the right to write into an existing object  $o$  if  $s$  does not do any read access. Hence, we cannot specify a simulation relation between the states of the two models that satisfies the properties needed, which is illustrated by the fact that a consistent RBAC state, where no access can be authorized, does not match any BLP state.

**Comparison mechanism** In [18], a similar approach has been used to compare the Chinese Wall (CW) and the BLP models. From a methodological point of view, these comparisons can be generalized by the following approach. Indeed, proving that  $\mathbb{M}_1[\rho_1] \ll \mathbb{M}_2[\rho_2]$  first consists in building an intermediate model  $\mathbb{M}_{12}[\rho_2]$  which is just a translation of  $\mathbb{M}_1[\rho_1]$  expressed in the formalism of  $\mathbb{M}_2[\rho_2]$ . This translation leads to interpret the security parameter  $\rho_1$  by a security parameter  $\rho_2 = \kappa_{\rho}(\rho_1)$ , and to define a relation between states of the two models. Then, a model  $\mathbb{M}_{12}[\kappa_{\rho}(\rho_1)]$  is obtained by defining a security predicate  $\Omega_{12}$  over  $\Sigma_2$  that corresponds to the predicate  $\Omega_1$  over  $\Sigma_1$ . From this translation, we get a simulation relation allowing to prove that  $\mathbb{M}_1[\rho_1] \ll \mathbb{M}_{12}[\kappa_{\rho}(\rho_1)]$ . Then, in a second step, it suffices to prove that each state satisfying the predicate  $\Omega_{12}$  also satisfies the predicate  $\Omega_2$ . Hence, in this way, we prove that for all  $\rho_1$ , there exists a parameter  $\rho_2$  such that  $\mathbb{M}_1[\rho_1] \ll \mathbb{M}_2[\rho_2]$ . Conversely, in order to prove that a model  $\mathbb{M}_1[\rho_1]$  is not more restrictive than a model  $\mathbb{M}_2[\rho_2]$ , we prove that there exists a parameter  $\rho_1$ , such that for all  $\rho_2$ ,  $\mathbb{M}_1[\rho_1] \not\ll \mathbb{M}_2[\rho_2]$ .

## 4. Conclusion

In the literature on access control, one can find papers presenting a particular access control mechanism through examples without any formalization (or generalization) of the concepts involved in the model. Of course, such papers are very useful to understand how a particular access control works but they provide little help to implement it. Hence, we need to define a generic formal framework in which many access control models could be specified, implemented and proved correct according to some security properties. In this paper, we have introduced a uniform way to specify and to define access control models and their implementations at several levels of specification. Such a formalism also allows to reason over security models and to compare them. We have defined a way to compare access control models based on the notion of simulation of implementations. The framework we have designed in this paper provides semantical tools for studying access control models, and, while the model seems to be fairly complicated, such complexity is needed to tackle the kind of questions addressed here. This framework has been successfully used to specify, to implement and to compare classical access control models (mandatory models such as the BLP model, the CW model, and the RBAC model).

Of course, the problem of considering access control policies at an abstract level in order to compare or to simulate them, or just to reason about them, has also been addressed in the literature. In [5], the authors use a state transition approach based on simulation to compare several discretionary access control mechanisms: they compare access control lists mechanism with capability based systems [16] (both the Lampson matrix capabilities mechanism [14, 9] and the capabilities as references model). They also compare in the same way the access control lists mechanism and the trust management approach [3]. Our framework share some concepts with the approach presented in [5] but it is rather different concerning the specification of requests and implementations. A similar approach can also be found in [1], where the functionality of simple RBAC models is compared to access control lists mechanisms: the equivalence of these two models in specifying access control policies is shown. Another approach can be found in [24] where a formal framework is defined in order to compare the expressive power of access control models and is applied to compare the RBAC model to discretionary access control and trust management approaches. More recently, in [25], access control policies viewed as composition of policy fragments and combinators allowing to create a single policy from "sub-policies" are studied: a semantics for such combinators is defined and used to compare, in terms of reasonability properties, languages for specifying access control policies. As a future work, our aim is to express

all these approaches in our framework in order to compare them. Of course, such a development would lead us to enrich our framework.

Now, we would like to extend our framework by considering several directions. First, our approach to compare access control models requires that the models share the same set of requests, and it seems desirable to relax this constraint. Indeed, the comparisons we have done over classical access control models only consider requests that consist in adding or removing accesses. In order to consider “administrative requests” allowing to change security functions of a state, we have now to extend our definitions in order to characterize the “semantical modifications” done by “administrative requests” over states of the system and to compare the expressive power of such requests for each model. This leads to consider the notion of weak-simulation of implementations when defining a preorder over access control models. Furthermore, we think that our work could be a first step to consider composition of policies. Indeed, in practice, an access of a subject over an object is often performed by applying several policies. For example, a user in a company has to access his office in a building according to an access control policy. He then has to conform to another access control policy in order to get access to the data on his computer. It could be interesting to study and to formalize several ways to compose access control models, to compare them and to express in a mathematical setting the properties these mechanisms of composition have to satisfy.

**Acknowledgements** Many thanks to Thérèse Hardin for enlightening discussions on this subject. This work was partially supported by the project HTTS funded by the Macao Science and Technology Development Fund and by the french SSURF ANR project ANR-06-SETI-016.

## References

- [1] J. Barkley. Comparing simple role based access control models and access control lists. In *ACM Workshop on Role-Based Access Control*, pages 127–132, 1997.
- [2] D. Bell and L. LaPadula. Secure Computer Systems: a Mathematical Model. Technical Report MTR-2547 (Vol. II), MITRE Corp., Bedford, MA, May 1973.
- [3] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management. In *Proc. of the 17th IEEE Computer Society Symposium on Research in Security and Privacy*, 1996.
- [4] D. F. C. Brewer and M. J. Nash. The chinese wall security policy. In *Proc. IEEE Symposium on Security and Privacy*, pages 206–214, 1989.
- [5] A. Chander, J. Mitchell, and D. Dean. A state-transition model of trust management and access control. In *Proc. of the 14th IEEE Computer Security Foundations Workshop CSFW*, pages 27–43. IEEE Computer Society Press, 2001.
- [6] D. F. Ferraiolo and D. R. Kuhn. Role-based access control. In *Proceedings of the 15th National Computer Security Conference*, 1992.
- [7] E. Gureghian, T. Hardin, and M. Jaume. A full formalisation of the Bell and Lapadula security model. Technical Report 2003-007, Univ. Paris 6, LIP6, 2003.
- [8] L. Habib. Formalisation, comparaison et implantation d’un modèle de contrôle d’accès à base de rôles. Master’s thesis, UPMC, Paris, France, 2007.
- [9] M. Harrison, W. Ruzzo, and J. Ullman. Protection in operating systems. *Communications of the ACM*, 19:461–471, 1976.
- [10] M. Jaume and C. Morisset. Formalisation and implementation of access control models. In *ITCC - IAS’05*, pages 703–708. IEEE CS Press, 2005.
- [11] M. Jaume and C. Morisset. A formal approach to implement access control. *Journal of Information Assurance and Security*, 2:137–148, 2006.
- [12] M. Jaume and C. Morisset. Towards a formal specification of access control. In *Workshop FCS-ARSPA’06*, pages 213–232, 2006.
- [13] M. Jaume and C. Morisset. On specifying, implementing and comparing access control models. A Semantical Framework. Technical report, Univ. Paris 6, LIP6, 2007.
- [14] B. Lampson. Protection. *Operating Systems Review*, 8(1):18–24, Jan. 1974.
- [15] L. LaPadula and D. Bell. Secure Computer Systems: A Mathematical Model. *Journal of Computer Security*, 4:239–263, 1996.
- [16] H. Levy. *Capability-Based Computer Systems*. Digital Press, Bedford, MA, 1984.
- [17] McLean. The algebra of security. In *Proc. IEEE Symposium on Security and Privacy*, pages 2–7. IEEE Computer Society Press, 1988.
- [18] C. Morisset. *Sémantique des systèmes de contrôle d’accès*. PhD thesis, Université Pierre et Marie Curie - Paris 6, 2007.
- [19] M. Nyanchama and S. L. Osborn. Modeling mandatory access control in role-based security systems. In *DBSec*, pages 129–144, 1995.
- [20] S. L. Osborn, R. S. Sandhu, and Q. Munawar. Configuring role-based access control to enforce mandatory and discretionary access control policies. *ACM Transactions on Information and System Security*, 3(2):85–106, 2000.
- [21] F. project. *Focal, version 0.2 Tutorial and reference manual*. LIP6 – INRIA – CNAM, sept 2004. Distribution available at: <http://focal.inria.fr>.
- [22] R. S. Sandhu. A lattice interpretation of the chinese wall policy. In *Proc. 15th NIST-NCSC National Computer Security Conference*, pages 329–339, 1992.
- [23] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, 1996.
- [24] M. Tripunitara and N. Li. Comparing the expressive power of access control models. In *11th ACM Conf. on Computer and Communications Security*. ACM SIGSAC, 2004.
- [25] M. Tschantz and S. Krishnamurthi. Towards reasonability properties for access-control policy languages. In D. Ferraiolo and I. Ray, editors, *Proceedings of SACMAT 2006*, pages 160–169. ACM, 2006.