# Rewrite Specifications of Access Control Policies in Distributed Environments

C. Bertolissi and M. Fernández

LIF, Marseille & King's College London

WTS'2010, Nancy

# Authorisation models and policies

Access control is of fundamental importance in computer security. Formal specifications of access control models and policies make it possible to

- compare policies rigorously,
- understand the consequences of modifying policies, and
- prove properties of policies.

# Context

Over the last few years, *a wide range of access control models* have been developed :
ACL, DAC, MAC, RBAC, EBAC, ...

In contrast, recently a *general meta-model* for access control based on the primitive notion of category has been proposed [Sacmat09].

# Authorisation models and policies

The meta-model approach has advantages :

- identify a core set of principles of access control, which can be specialised for domain-specific applications.
- abstract away many of the complexities that are found in specific access control models ;
- help to simplify the task of policy writing.

# Motivations

We propose a formal specification of Barker's meta-model using term rewriting.
This choice has several motivations :

- *Expressivity* : rewriting systems have been used to specify, in a uniform way, various computational paradigms.
- *A well-developed theory* : rewriting techniques can be used to prove properties of policies specified as rewriting systems.
- *Availability of tools* such as ELAN, MAUDE, CiME, etc. for rapid prototyping of access policies.

# Contributions

- a declarative, rewrite-based specification of the category-based access control model, together with a formal operational semantics.
- a technique to prove totality and consistency of access control policies.
- the encoding of well-known access control models (H-RBAC, MAC, DAC and DEBAC models) in the meta-model, to demonstrate its expressive power.
- the axiomatisation of the meta-model for taking into account the requirements of distributed systems, together with a rewrite-based operational semantics.

# Overview

- The Category-based meta-model $\mathcal{M}$.
- Introduction to Term rewriting.
- Rewrite-based specification of $\mathcal{M}$ :
  - definition,
  - request evaluation,
  - properties
  - expressive power.
- The Distributed version of $\mathcal{M}$.
- Conclusions and future work

# The meta-model $\mathcal{M}$ : notion of category

A key aspect the meta-model is the notion of a category.

A category is a class, a domain to which entities or concepts belong. We regard categories as a primitive concept.

Classic types of groupings used in access control, like a *role*, a *security clearance*, a discrete *measure of trust*, etc, are particular instances of the notion of category.

# Features of the AC meta-model

Entities in the meta-model $\mathcal{M}$ are denoted uniquely by constants in a many sorted domain of discourse, including :

- A countable set $\mathcal{C}$ of *categories* : $c_0, c_1, \ldots$.
- A countable set $\mathcal{P}$ of *principal identifiers* : $p_1, p_2, \ldots$.
- A countable set $\mathcal{A}$ of named *actions* : $a_1, a_2, \ldots$.
- A countable set $\mathcal{R}$ of *resources* : $r_1, r_2, \ldots$.
- A finite set $\mathcal{A}uth$ of *answers* : *e.g.* grant, deny.

Additionally we may have :

- A countable set $\mathcal{S}$ of *situational identifiers* (locations, system states,...) : $s_0, s_1, \ldots$.
- A countable set $\mathcal{E}$ of *event identifiers* : $e_1, e_2, \ldots$.
- A countable set $\mathcal{T}$ of *time points*.

Entities are assigned to distinct classes or groups called categories.

# Relationships between entities

- *Principal-category assignment* $\mathcal{PCA}$ : $(p, c) \in \mathcal{PCA}$ iff a principal $p \in \mathcal{P}$ is assigned to the category $c \in \mathcal{C}$.
- *Permissions* $\mathcal{ARCA}$ : $(a, r, c) \in \mathcal{ARCA}$ iff the action $a \in \mathcal{A}$ on resource $r \in \mathcal{R}$ can be performed by principals assigned to the category $c \in \mathcal{C}$.
- *Authorisations* $\mathcal{PAR}$ : $(p, a, r) \in \mathcal{PAR}$ iff a principal $p \in \mathcal{P}$ can perform the action $a \in \mathcal{A}$ on the resource $r \in \mathcal{R}$.
- *Banned actions on resources* $\mathcal{BARCA}$ : $(a, r, c) \in \mathcal{BARCA}$ iff the action $a \in \mathcal{A}$ on resource $r \in \mathcal{R}$ is forbidden for principals assigned to the category $c \in \mathcal{C}$.
- *Barred access* $\mathcal{BAR}$ : $(p, a, r) \in \mathcal{BAR}$ iff performing the action $a \in \mathcal{A}$ on the resource $r \in \mathcal{R}$ is forbidden for the principal $p \in \mathcal{P}$.

# Distributed version of $\mathcal{M}$

A distributed system is generally composed of several sites, with different policies in place at each site.

We consider families of relations, e.g $\mathcal{BAR}_s$, $\mathcal{PAR}_s$, indexed by situational identifiers (i.e., sites).

The relation $\mathcal{PAR}$ defining the global authorisation policy will be obtained by composing the local policies defined by the relations $\mathcal{PAR}_s$ and $\mathcal{BAR}_s$.

# Axioms of Distributed $\mathcal{M}$

In any site $s$ of the distributed system, the following axioms hold :

$(b1)$ $\forall p \in \mathcal{P},\ \forall a \in \mathcal{A},\ \forall r \in \mathcal{R},\ \forall c \in \mathcal{C},\ \forall s \in \mathcal{S}$
$(p, c) \in \mathcal{PCA}_s \wedge (\exists c' \in \mathcal{C}, c \subseteq c' \ \wedge (a, r, c') \in \mathcal{ARCA}_s) \Rightarrow (p, a, r) \in \mathcal{PAR}_s$

$(c1)$ $\forall p \in \mathcal{P},\ \forall a \in \mathcal{A},\ \forall r \in \mathcal{R},\ \forall c \in \mathcal{C},\ \forall s \in \mathcal{S}$
$(p, c) \in \mathcal{PCA}_s \wedge (\exists c' \in \mathcal{C}, c' \subseteq c \ \wedge (a, r, c') \in \mathcal{BARCA}_s) \Rightarrow (p, a, r) \in \mathcal{BAR}_s$

$(d1)$ $\forall p \in \mathcal{P},\ \forall a \in \mathcal{A},\ \forall r \in \mathcal{R},\ \forall c \in \mathcal{C},\ \forall s \in \mathcal{S}$
$(p, c) \in \mathcal{PCA}_s \ \wedge \ (a, r, c) \notin \mathcal{ARCA}_s \ \wedge \ (a, r, c) \notin \mathcal{BARCA}_s \Rightarrow$
$(p, a, r) \in \mathcal{UNDET}_s$

$(e1)$ $\forall s \in \mathcal{S},\ \ \mathcal{ARCA}_s \cap \mathcal{BARCA}_s = \emptyset$

# Axioms of Distributed $\mathcal{M}$

The axioms below describe the global authorisation relation in terms of the local policies defined at each site :

$$(f1) \quad \forall p \in \mathcal{P},\ \forall a \in \mathcal{A},\ \forall r \in \mathcal{R},\\ (p,a,r) \in \mathcal{OP}_{par}(\{\mathcal{PAR}_s, \mathcal{BAR}_s \mid s \in S\}) \Rightarrow (p,a,r) \in \mathcal{PAR}$$

$$(g1) \quad \forall p \in \mathcal{P},\ \forall a \in \mathcal{A},\ \forall r \in \mathcal{R},\\ (p,a,r) \in \mathcal{OP}_{bar}(\{\mathcal{PAR}_s, \mathcal{BAR}_s \mid s \in S\}) \Rightarrow (p,a,r) \in \mathcal{BAR}$$

$$(h1) \quad \mathcal{PAR} \cap \mathcal{BAR} = \emptyset$$

The final authorisation is computed specialising the definition of the operators $\mathcal{OP}_{par}$ and $\mathcal{OP}_{bar}$. For example,

- $\mathcal{OP}_{bar} = (\mathcal{BAR}_s \vee \mathcal{BAR}_t)$ and $\mathcal{OP}_{par} = ((\mathcal{PAR}_s / \mathcal{BAR}_t) \vee (\mathcal{PAR}_t / \mathcal{BAR}_s))$ specifies a combination giving priority to deny
- $\mathcal{OP}_{par} = \mathcal{PAR}_\varsigma$ and $\mathcal{OP}_{bar} = \mathcal{BAR}_\varsigma$ if the system has a central policy at site $\varsigma$

# Operational semantics

The formal specification of the operational semantics of the meta-model is given using *term rewriting*.

This choice has several motivations :

- Expressivity : rewriting systems have been used to specify, in a uniform way, various computational paradigms.
- A well-developed theory : rewriting techniques can be used to prove properties of policies specified as rewriting systems.
- Availability of tools such as ELAN, MAUDE, CiME, etc. for rapid prototyping of access policies.

# Term Rewrite systems

Term rewrite systems (TRSs) are defined by a set of terms and a set of rewrite rules that are used to 'reduce' terms.

The set of terms $T(\mathcal{F}, \mathcal{X})$ is built up from a signature $\mathcal{F}$ and a set of variables $\mathcal{X}$.

The set of rewrite rules is of the form $R = \{l_i \to r_i\}_{i \in I}$ where $l_i, r_i \in T(\mathcal{F}, \mathcal{X})$, $l_i \notin \mathcal{X}$, and $Var(r_i) \subseteq Var(l_i)$.

We denote a rewrite step by $t \longrightarrow t'$ and its reflexive transitive closure by $t \longrightarrow^* t'$.

If a term $t$ cannot be reduced further, we say $t$ is in *normal form*.

# Example : lists of naturals

$$\mathcal{F} = \{0, s\} \cup \{nil, cons, \text{Length}\}$$

$$\mathcal{T} = (\mathcal{F}, \{x, y, \ldots\})$$

$$\mathcal{R} = \begin{cases} R_0 : & \text{Length(nil)} & \rightarrow & 0 \\ R_1 : & \text{Length(cons}(x, l)) & \rightarrow & \text{S(Length}(l)) \end{cases}$$

Term reduction sequence :

$$\begin{aligned} & \text{Length(cons}(0, \text{cons}(\text{S}(0), \text{nil}))) \\ \rightarrow_{R_1} \quad & \text{S(Length(cons}(\text{S}(0), \text{nil})) \\ \rightarrow_{R_1} \quad & \text{S(S(Length(nil)))} \rightarrow_{R_0} \text{S(S}(0)) \end{aligned}$$

# Distributed Rewriting

Distributed term rewrite systems (DTRSs) are TRSs where rules are partitioned into *modules*, each associated with a unique identifier, and function symbols are annotated with such identifiers.

In a DTRS, we can associate a module to each site of a distributed system : we may write $f_\nu$ to refer to the definition of the function symbol $f$ in the site $\nu$.

We assume that each module has a unique identifier ;
If a symbol is used in a rule without a site annotation, we assume the function is defined locally.

# Rewrite-based specification

The rewrite-based specification of the axioms of Distributed $\mathcal{M}$ is as follows :

| | | |
|---|---|---|
| *Pca* | pca(p) | $\longrightarrow$ [c] |
| *Arca* | arca(c) | $\longrightarrow [(a_1, r_1), \ldots, (a_k, r_k)]$ |
| *Barca* | barca(c) | $\longrightarrow [(a_l, r_l), \ldots, (a_t, r_t)]$ |
| *Contain* | contain(c) | $\longrightarrow [c, c_1, \ldots, c_n]$ |
| $Par_s$ | par($P, A, R$) | $\longrightarrow$ *if* $(A, R) \in$ arca$^*$(contain(pca($P$))) *then* grant *else* *if* $(A, R) \in$ barca$^*$(contain(pca($P$))) *then* deny *else* undet |
| *Arca*$^*$ | arca$^*$(cons($C, L$)) | $\longrightarrow$ append(arca($C$), arca$^*$($L$)) |
| *Arca*$^*$ | arca$^*$(cons($C$, nil)) | $\longrightarrow$ nil |
| *Barca*$^*$ | barca$^*$(cons($C, L$)) | $\longrightarrow$ append(barca($C$), barca$^*$($L$)) |
| *Barca*$^*$ | barca$^*$(cons($C$, nil)) | $\longrightarrow$ nil |

# Rewrite-based specification

Global authorisations are computed using the following rewrite rules (which implement axioms f1 and g1) :

$$\text{Authorised}(P, A, R, s_1, \ldots, s_n) \rightarrow \text{fauth}(op, \text{par}_{s_1}(p, a, r), \ldots, \text{par}_{s_n}(p, a, r))$$

$$\text{fauth}(op, \text{par}_{s_1}(p, a, r), \ldots, \text{par}_{s_n}(p, a, r)) \rightarrow \text{answ} \quad \textit{with} \ \text{answ} \in \mathcal{A}\text{uth}$$

where the function fauth combines the results into a final access authorisation according to the operator *op*.
(e.g. union, $\text{fauth}(\textit{union}, \text{deny}, x) \rightarrow \text{deny}$,
but more sophisticated combinations are possible).

# Evaluating access requests

An access request by a principal $p$ to perform the action $a$ on the resource $r$ can then be evaluated simply by rewriting the term Authorised$(p, a, r, s_i)$ to normal form.

*Proposition*

The given rewrite-based definition is a correct realisation of the axioms (f1) and (g1) :

Authorised$(p, a, r, s_i) \rightarrow^*$ grant (respectively deny) if and only if $(p, a, r) \in \mathcal{PAR}$ (respectively $(p, a, r) \in \mathcal{BAR}$).

# Expressiveness of the meta-model

A range of existing access control models can be represented as specialised instances of our meta-model [Essos'10] :

- (static) access control models, such as DAC and MAC (including the well-known Bell-LaPadula model),
- RBAC (including temporal and location constraints),
- the Chinese Wall policy,
- as well as dynamic models, such as DEBAC.

# Combining RBAC and Bell-Lapadula policies

Consider a principal $p$ working in an organisation where employees share an electronic agenda $a$, maintained on a server $\nu$.

This organisation adopts an RBAC policy for the employees, and moreover uses a specific Bell-Lapadula policy on the agenda in site $\nu$.

The RBAC policy provides rules such as

$$\begin{aligned}
&\mathsf{pca}_\pi(p) \longrightarrow [\mathsf{employee}]\\
&\mathsf{arca}_\pi(\mathsf{employee}) \longrightarrow [(\mathsf{read}, \mathsf{reportA}), (\mathsf{write}, \mathsf{a_{all}}), (\mathsf{read}, \mathsf{a_{all}}]\\
&\mathsf{barca}_\pi(\mathsf{employee}) \longrightarrow [(\mathsf{write}, \mathsf{reportA}), \ldots, ]
\end{aligned}$$

with $\pi$ the site where the principal is registered.

# Combining RBAC and Bell-Lapadula policies

We have in addition a Bell-Lapadula policy local to site $\nu$. The privileges depend on the secrecy level :

$\text{arca}_\nu(\text{top\_secret}) \rightarrow [(\text{read}, a_{ts}), (\text{write}, a_{ts}), (\text{read}, a_s), (\text{read}, a_p), ]$
$\text{barca}_\nu(\text{top\_secret}) \rightarrow [(\text{write}, a_s), (\text{write}, a_p)]$
...
$\text{arca}_\nu(\text{public}) \rightarrow [(\text{write}, a_p), (\text{read}, a_p), ]$
$\text{barca}_\nu(\text{public}) \rightarrow [(\text{write}, a_s), (\text{write}, a_{ts}), (\text{read}, a_s), (\text{read}, a_{ts})]$

Assume the princiapl p is assigned to the public level,
$\text{pca}_\nu(p) \rightarrow [\text{public}]$.

# Combining RBAC and Bell-Lapadula policies

Consider the request of editing a section $a_s$ in the agenda by the principal *p*.

The request evaluation starts by calling the Authorised function local to the site where the request is issued

$$\begin{aligned} &\text{Authorised}(p, \text{write}, a_s, \pi, \nu) \\ &\rightarrow^* \text{fauth}(\text{union}, \text{par}_\pi(p, \text{write}, a_s), \text{par}_\nu(p, \text{write}, a_s)) \end{aligned}$$

We evaluate the access authorisation in site $\pi$ (with an RBAC policy), and also in site $\nu$ (with a Bell-Lapadula policy) using a union operator.

Access will be permitted if both policies return a grant answer.

# Combining RBAC and Bell-Lapadula policies

We consider the evaluation of the request in the site $\pi$ :

$\mathsf{par}_{\pi}(\mathsf{p}, \mathsf{write}, \mathsf{a_s}) \rightarrow^{*} \mathsf{grant}$

since p is an employee of the organisation.

We consider now the evaluation of the request in site $\nu$ :

$\mathsf{par}_{\nu}(\mathsf{p}, \mathsf{write}, \mathsf{a_s}) \rightarrow^{*} \mathsf{deny}$

since *p* is assigned to the public level in the Bell-Lapadula policy.

Thus finally the access will be denied

$$\mathsf{Authorised}(\mathsf{p}, \mathsf{write}, \mathsf{a_s}, \pi, \nu) \rightarrow^{*}$$
$$\mathsf{fauth}(\mathsf{union}, \mathsf{grant}, \mathsf{deny}) \rightarrow^{*} \mathsf{deny}$$

# Properties of the policy

*Totality* : Each request from a valid principal $p$ to perform an action $a$ on the resource $r$ receives as answer : a permission or a denial.

*Consistency* : For any $p \in \mathcal{P}$, $a \in \mathcal{A}$, $r \in \mathcal{R}$, at most one result is possible for an authorisation request Authorised$(p, a, r)$. In other words, at most one of the results grant, deny, undetis possible for each request.

*Soundness and Completeness* : For any $p \in \mathcal{P}$, $a \in \mathcal{A}$, $r \in \mathcal{R}$, an access request by $p$ to perform the action $a$ on $r$ is granted if and only if $p$ belongs to a category that has the permission $(a, r)$.

# Properties of the policy

Totality and consistency can be ensured, for policies defined as term rewriting systems, by checking that the rewrite relation is

- *confluent*, that ensures that results are unique.
- *terminating*, that ensures that all requests produce a result (their evaluation cannot get “stuck”).

There are several results that provide sufficient conditions for these properties to hold.
[Newmann,Huet,Klop90,Baader-Nipkow98,Jouannaud-Okada97,...].
For example we can use ortogonality, absence of critical pairs, hierarchical term rewriting, . . .

# Conclusion

We have given a rewrite-based specification of a distributed meta-model of access control that is based on common, core concepts of access control models.

The term rewriting approach can be used to give a meaningful uniform semantics to policies, facilitates the task of proving properties of policies and provide an executable specification of the policy.

In future work, we will investigate an algebra for combination operators, the design of languages for policy specification and the practical implementation of category-based policies.