Partial model checking, process algebra operators and satisfiability procedures for (automatically) enforcing security properties

Fabio Martinelli, Ilaria Matteucci

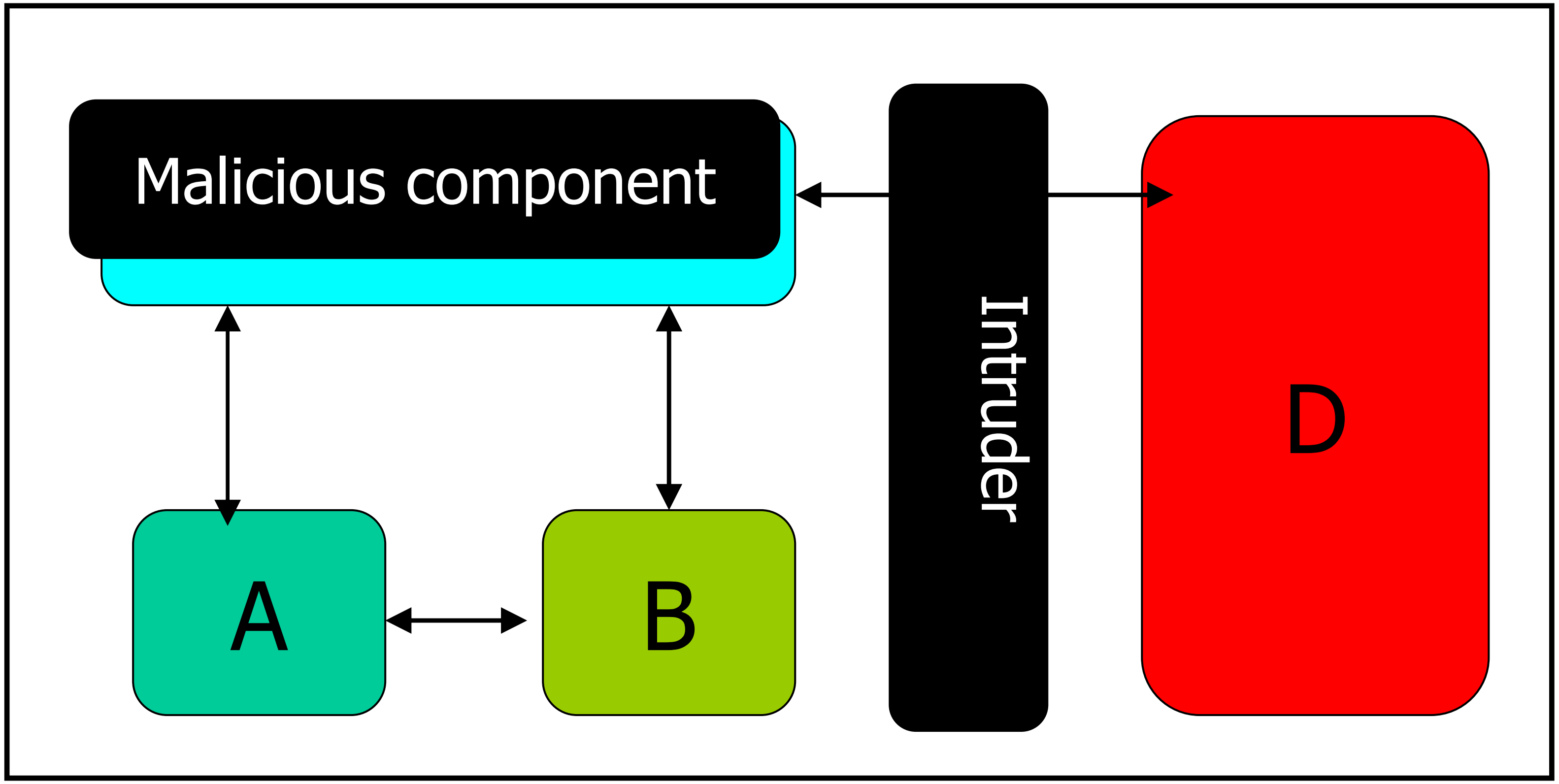
Istituto di Informatica e Telematica
Consiglio Nazionale delle Ricerche
IIT-CNR, Pisa, Italy

FCS,30 June -1 July 2005

Outline

- Open systems for security analysis
  - Logical approach
  - Non-interference
- Partial model checking
  - Dealing with information flow properties: (B)NDC
- Controller operator
  - Definition
  - How to use it
- Synthesis
- Other controllers
- Conclusion

Security analysis as open systems analysis



Specification: A | B | [ ] | D | [ ]

Open system verification

An open system **S(_)** satisfy a property ϕ iff:

For all X we have $\quad$ **S|X** $\models \phi$

Where ϕ is a logic formula.

X is the unknown entity whose behavior cannot be predicted but whose presence must be considered.

Partial model checking (Andersen '95)

- Given a (finite) system **S**, and a formula ϕ, then we can compute a formula $\phi_{//S}$ s.t.:

$$S | X \models \phi$$

iff

$$X \models \phi_{//S}$$

  - This is called **partial model checking (PMC)** since the behavior of the whole system, i.e. **S|X**, is only partially evaluated.

PMC for dealing with universal quantification

The presence of universal quantification makes it difficult to check open systems properties:

For all X we have $S | X \models \phi$

It would be easier to verify:

For all X we have $X \models \phi_{//S}$

Which is a validity checking problem of a logic formula.

Through PMC, we can perform a similar reduction.

How PMC works ..

Assume to have a language where the unique operator is:

$$\frac{A \xrightarrow{1} \qquad B \xrightarrow{2}}{A|B \xrightarrow{3}}$$

Assume to have **S** s.t. $\mathbf{S} \xrightarrow{1}$ and consider the formula $\exists \mathcal{X}^{\mathbf{3}}$ says "the process may perform the action 3" then:

$\mathbf{S}|\mathbf{X} \models \exists \mathcal{X}^{\mathbf{3}}$ iff (see the semantics rule)

$\mathbf{S} \xrightarrow{1}$ **and** $\mathbf{X} \xrightarrow{2}$ iff (see the actions of S)

$\mathbf{X} \xrightarrow{2}$ iff

$\mathbf{X} \models \exists \mathcal{X}^{\mathbf{2}}$ "the process may perform the action 2"

Our problem

We use a logical approach to describe a **non-interference** property (Martinelli '98) :

There are two users *High* and *Low* interacting with the same computer system. We ask if there is any **flow of information** from *High* to *Low*.

We denote with *BNDC* a security property (Focardi-Gorrieri '94) s.t.:

For all high users X we have $(S|X)\backslash H \approx S\backslash H$

May be reduced to a verification problem for open system trough the use of characteristic formulae

For all high users X we have $(S|X)\backslash H \models \phi^{\approx S\backslash H}$

PMC for BNDC analysis

- Through **partial model checking** we can reduce the *BNDC* checking to a validity check for logic as follows:

For all high users X we have $(S|X)\backslash H \models \phi^{\approx S\backslash H}$

iff

For all high users X we have $X \models (\phi^{\approx S\backslash H})_{//S\backslash H}$

- The validity checking problem is decidable for the logic used to express the characteristic formulae. Thus, we obtain a decidability result about the *BNDC* verification for finite systems

If the security property is not satisfied?

We may simply check each processes **X** before executing it or, if we do not have this possibility, we may define a **controller** that in any case force it to behave correctly.

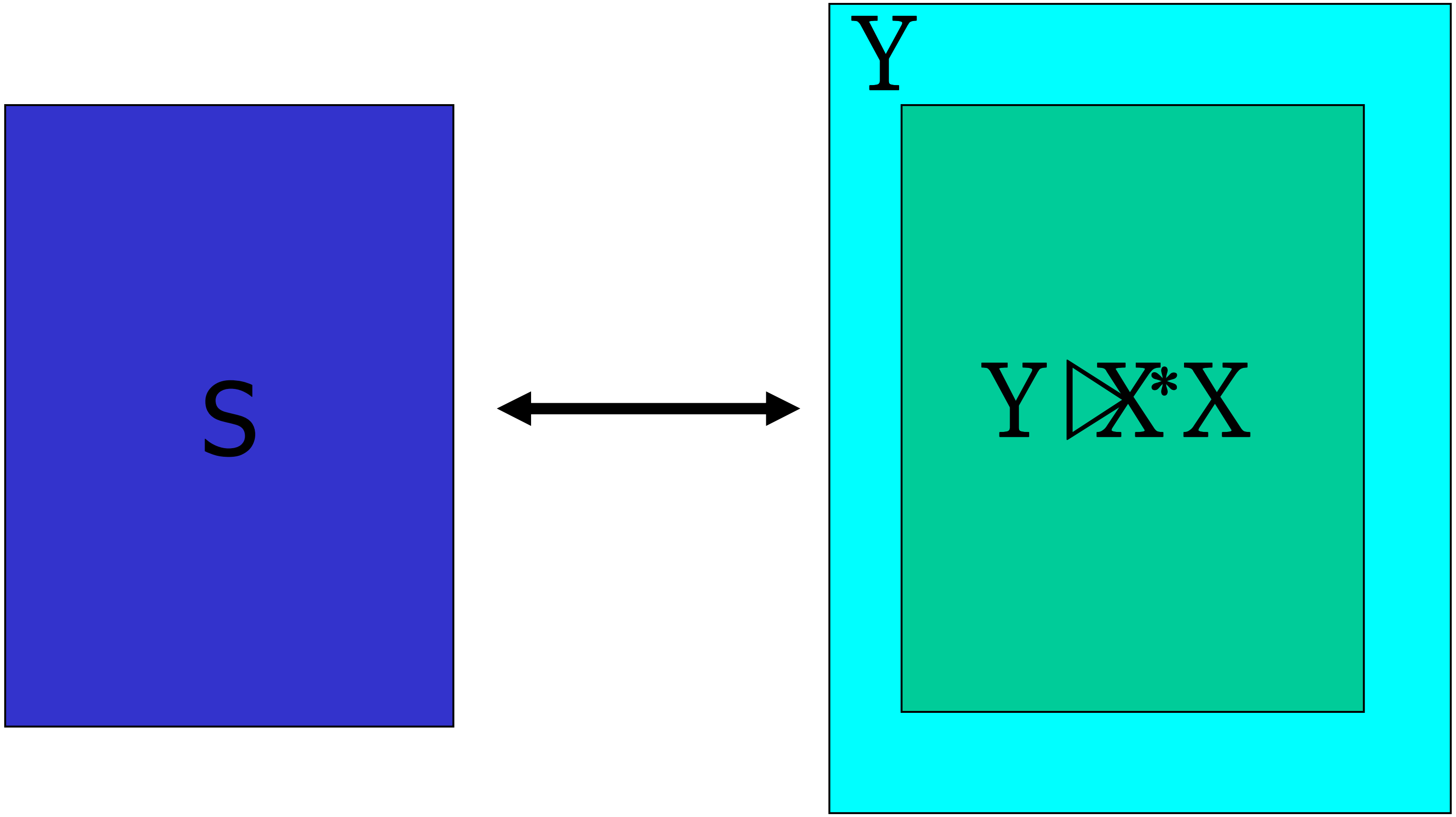
Enforcing security properties: a controller operator

In order to enforce specific security properties a new operator, said **Y** $\triangleright^*$ **X**, is defined. It can permit to control the behavior of the component **X**, given the behavior of a control program **Y**.

Esempio:

$$\frac{E \xrightarrow{a} E' \quad F \xrightarrow{a} F'}{E \triangleright' F \xrightarrow{a} E' \triangleright' F'} \qquad \frac{E \xrightarrow{a} E'}{E \triangleright' F \xrightarrow{a} E' \triangleright' F}$$

Controller operator $\triangleright^*$



Specification: $S|(Y \triangleright^* X)$

Our solution (1)

A system **S|(Y $\triangleright^*$ X)** always enjoys the desired security properties even if **X** tries to break the security property. Thus, a control program **Y** is s.t.:

For all X we have (S |(Y $\triangleright^*$ X))\H $\models \phi$

Equivalently, by **partial model checking** we get:

$$\exists \mathbf{Y} \, \forall \mathbf{X} \, (\mathbf{Y} \triangleright^* \mathbf{X}) \models \phi_{//\, \mathbf{S} \backslash \mathbf{H}}$$

Our solution (2)

For every **X** and **Y**, if we have:

$$\mathbf{Y} \triangleright^{*} \mathbf{X} \sim \mathbf{Y}$$

Then

$$\exists\, \mathbf{Y}\, \forall\, \mathbf{X}\, (\mathbf{Y} \triangleright^{*} \mathbf{X}) \models \phi_{//\, \mathbf{S}/\mathbf{H}} \qquad \textbf{(1)}$$

becomes

$$\exists\, \mathbf{Y} \text{ s.t. } \mathbf{Y} \models \phi_{//\, \mathbf{S}/\mathbf{H}} \qquad \textbf{(2)}$$

An example:

In order to verify that both of these processes satisfy *BNDC*, it is sufficient that **Y** $\triangleright^{*}$ **X** and **Y** are weakly bisimilar.

Synthesis of the program controller

It is possible to find a program controller **Y** like in **(2)**, that is model of $\phi_{//\, S/H}$.

<u>We use the well - known results on satisfiability</u>

Given a formula ϕ it is possible to decide in exponential time in length of ϕ if there exists a model of ϕ and it is also possible to give an example of it.

Other controllers

1)

$$\frac{E \xrightarrow{a} E' \quad F \xrightarrow{a} F'}{E \triangleright'' F \xrightarrow{a} E' \triangleright'' F'} \qquad \frac{E \xrightarrow{a} E' \quad F \not\xrightarrow{a} F'}{E \triangleright'' F \xrightarrow{a} E' \triangleright'' F}$$

2)

Enforcing Monitor of Schneider

$$\frac{E \xrightarrow{a} E' \quad F \xrightarrow{a} F'}{E \triangleright''' F \xrightarrow{a} E' \triangleright''' F'}$$

A simple example (1)

Consider the process :

$$S = l.0 + h.h.l.0$$

S\h is weakly bisimilar to **l.0**.

Consider the following equational definition:

$$X_S =_\nu [\tau]X_S \wedge [l]\, T \wedge \langle\langle l \rangle\rangle\, T$$

After partial evaluation:

$$(X_S)_{//S} =_\nu [\,\tau\,](X_S)_{//S} \wedge [\hbar]\langle\langle \hbar \rangle\rangle T$$

A simple example (2)

Using $\triangleright''$, we find a model $(X_S)_{//S}$: $Y=\hbar.\hbar.0$

Then

$$\forall X \; (S \mid (Y \triangleright'' X))\backslash h \;\; \text{satisfies} \;\; (X_S)_{//S}$$

For instance, considering $X=\hbar.0$, the system becomes:

$$(S \mid (Y \triangleright'' X))\backslash h \xrightarrow{\tau} (h.l.0 \mid \hbar \triangleright'' 0)\backslash h$$

Thus

$$(h.l.0 \mid \hbar \triangleright'' 0)\backslash h \xrightarrow{\tau} (l.0 \mid 0 \triangleright'' 0)\backslash h \approx l.0$$

Conclusion and future work

- We contributed to extend a framework based on process calculi and logical techniques in order to model and verify several security properties.
  - A benefit of our logical approach is the usage of validity checking as verification and in order to find satisfiability procedures for enforcing security properties.
- We added also the possibility to automatically build enforcing mechanisms.
- Our approach could be make more feasible in practice. We are looking for security properties whose corresponding controllers may be built more efficiently.
- Our approach has been recently extended to cope with timed security properties.

Thank you all!!!

Three possible scenarios

We may distinguish several situations depending on the control

one may have on the process **X**:

1. if **X** performs an action we may detect and intercept it;
2. in addition to 1), it is possible to know which are the possible next steps of **X**;
3. if **X** whole code is known we are able to model check.

Bisimulation equivalence

Let *R* be a binary relation over a set of processes *E*. Then *R* is called **strong bisimulation** ($\sim$) if and only if, whenever (E,F) $\in$ *R* we have

- **If E $\xrightarrow{a}$ E' then $\exists$ F' s.t. F $\xrightarrow{a}$ F' and (E',F') $\in$ R**
- **If F $\xrightarrow{a}$ F' then $\exists$ E' s.t. E $\xrightarrow{a}$ E' and (F',E') $\in$ R**

The notion of observational relations is the follow:

E $\xrightarrow{\tau}$ **E'** (or **E** $\Rightarrow$ **E'**) if **E** $\xrightarrow{\tau}$***E'** for **a** $\neq \tau$, **E** $\overset{a}{\Rightarrow}$ **E'** if **E** $\overset{\tau}{\Rightarrow}\xrightarrow{a}\overset{\tau}{\Rightarrow}$ **E'.**

where τ is the internal action.

Let *R* be a binary relation over a set of process *E*. Then *R* is said to be a **weak bisimulation** ($\approx$) if, whenever (E, F) $\in$ *R*:

- **If E $\xrightarrow{a}$ E' then $\exists$ F' s.t. F $\overset{a}{\Rightarrow}$ F' and (E',F') $\in$ R**
- **If F $\xrightarrow{a}$ F' then $\exists$ E' s.t. E $\overset{a}{\Rightarrow}$ E' and (F',E') $\in$ R**

Process algebra (CCS) (Milner '89)

Process algebra (CCS) is used in order to specify a lot of kind of system.

Syntax of expression:

P ::= 0 | *A* | *a*.P | P+P | P||P | P/L | P[f]

Where 0 is deadlock, *A* is a set of name of processes (agents) and $a \in Act = \mathcal{L} \cup \overline{\mathcal{L}} \cup \tau$ where τ is an internal action.

Background about logic

- A logic usually consist of:
  - A set of formulae, e.g.:
    - F *and* F, F *or* F, F *implies* F, …..
  - A truth relation ⊨ between structures and formulae
    - S ⊨ F means that S is a model for F
    - F is valid, written ⊨ F, whenever S ⊨ F for every structure S
    - F is satisfiable if there exists S, S ⊨ F
  - A set of actions and rules. These induce a deduction relation ⊢ between formulae
    - F_1 .. F_n|-F means F can be p roved from F_1 , .., F_n through a sequence of applications of axioms and rules
    - We assume that if |-F then |= (soundness)

Equational μ-calculus

Let a be in Act and X be a variable

(Assertion)

$A ::= X \mid T \mid F \mid X_1 \wedge X_2 \mid X_1 \vee X_2 \mid \langle \alpha \rangle X \mid [\alpha] X$

(Equation)

$D ::= X =_{\nu} AD \mid X =_{\mu} AD \mid \varepsilon$

It is very suitable for partial model checking

Semantic of CCS

prefix $$\frac{}{\alpha.P \xrightarrow{\alpha} P}$$

choice $$\frac{P \xrightarrow{\alpha} P'}{P+Q \xrightarrow{\alpha} P'+Q} \quad \frac{Q \xrightarrow{\alpha} Q'}{P+Q \xrightarrow{\alpha} P+Q'}$$

parallel $$\frac{P \xrightarrow{\alpha} P'}{P|Q \xrightarrow{\alpha} P'|Q} \quad \frac{Q \xrightarrow{\alpha} Q'}{P|Q \xrightarrow{\alpha} P|Q'} \quad \frac{Q \xrightarrow{\alpha} Q' \quad P \xrightarrow{\bar{\alpha}} P'}{P|Q \xrightarrow{\tau} P'|Q'}$$

restriction $$\frac{P \xrightarrow{\alpha} P'}{P \backslash L \xrightarrow{\alpha} P' \backslash L} \alpha, \bar{\alpha} \notin L$$

relabeling $$\frac{P \xrightarrow{\alpha} P'}{P[f] \xrightarrow{f(\alpha)} P'[f]}$$

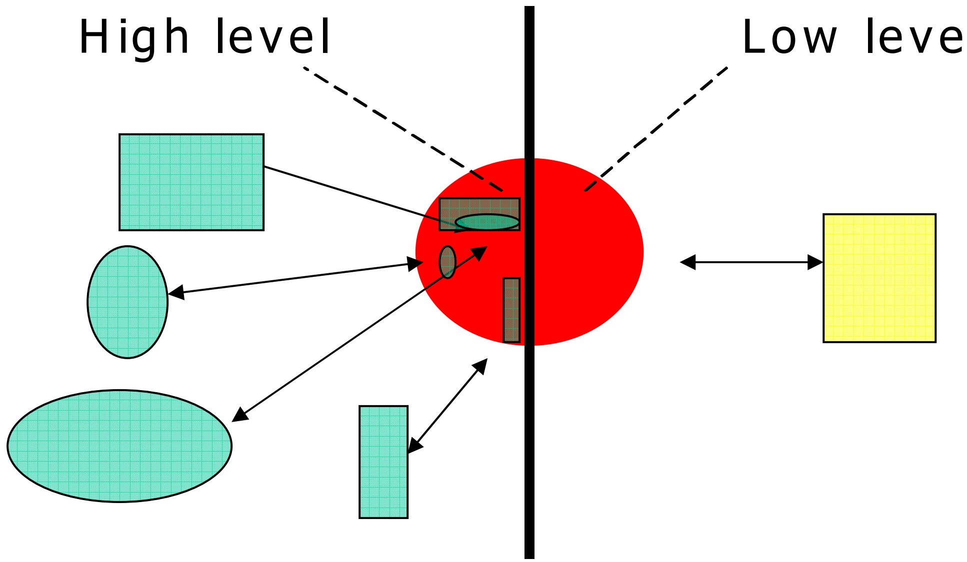
Characteristic formulas

- We can characterize through a formula the observational equivalence $\approx$
- Thus, given two (finite) systems **S** and $\mathbf{S_1}$, we can find a formula $\phi^{\approx S}$ s.t.:

$$S_1 \approx S \quad \textbf{iff} \quad S_1 \models \phi^{\approx S}$$

- Such characteristic formulas may be obtained for several system equivalences

System security properties: Non-interference (NI)



The system acts as an interface between high and low users. The high level activities must not interfere with the low level ones.