Verifying Persistent Security Properties

Annalisa Bossi, Damiano Macedonio, Riccardo Focardi,
Carla Piazza, and Sabina Rossi

Dipartimento di Informatica
Università Ca' Foscari di Venezia

{bossi,mace,focardi,piazza,srossi}@dsi.unive.it

Pisa, November 2003

Protect Confidential Data in a Multilevel System

- ▷ **Information Flow Security** aims at guaranteeing that no high level (confidential) information is revealed to users at low level, even in the presence of any possible malicious process
- ▷ **Non-Interference**: information does not flow from high to low if the high behavior has no effect on what low level can observe
- ▷ **Dynamicity**: a program which is in a secure state for a certain environment might become unprotected if the environment suddenly changes

Problem: incrementally build, rectify, and verify secure processes

Plan of the Talk

- ▷ The Security Process Algebra Language
- ▷ Information Flow Security as Unwinding Conditions
- ▷ Some instances: P_BNDC, SBNDC, CP_BNDC, PP_BNDC
- ▷ Incrementally Build secure processes
- ▷ Rectify non secure processes
- ▷ Verify security properties

The SPA syntax

E	$::=$	$\mathbf{0}$	*empty process*
	$\mid$	$a.E$	*input*
	$\mid$	$\bar{a}.E$	*output*
	$\mid$	$\tau.E$	*internal action*
	$\mid$	$E + E$	*non-det. choice*
	$\mid$	$E \mid E$	*parallel composition*
	$\mid$	$E \setminus v$	*restriction*
	$\mid$	$E[f]$	*relabelling*
	$\mid$	Z	*constant*

- H high actions and L low actions

The SPA semantics - Transitions

Semantics given through transition relations $\longrightarrow$ among processes defined by axioms and inference rules

Input $$\frac{}{a.E \xrightarrow{a} E}$$

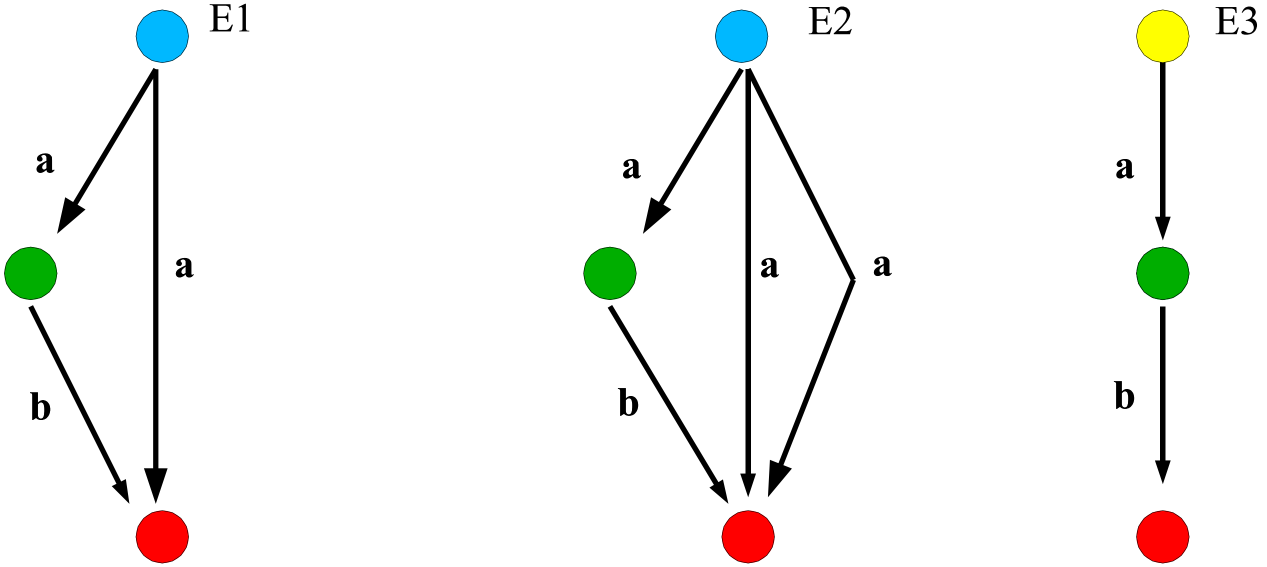
Output $$\frac{}{a.E \xrightarrow{\bar{a}} E}$$

Parallel $$\frac{E_1 \xrightarrow{a} E_1'}{E_1|E_2 \xrightarrow{a} E_1'|E_2} \qquad \frac{E_1 \xrightarrow{a} E_1' \quad E_2 \xrightarrow{\bar{a}} E_2'}{E_1|E_2 \xrightarrow{\tau} E_1'|E_2'}$$

Two processes are equivalent if they are weakly bisimilar: $E \approx_B F$

The SPA semantics - Bisimulation

- ▷ Idea: bisimulation is a mutual step-by-step simulation
- ▷ $E1 = a.b.\mathbf{0} + a.\mathbf{0} \qquad E2 = a.b.\mathbf{0} + a.\mathbf{0} + a.\mathbf{0} \qquad E3 = a.b.\mathbf{0}$



- ▷ $E1$ and $E2$ are bisimilar and they both simulate $E3$
- ▷ $E3$ can simulate the rightmost a of $E1$, but it is not bisimilar to $E1$

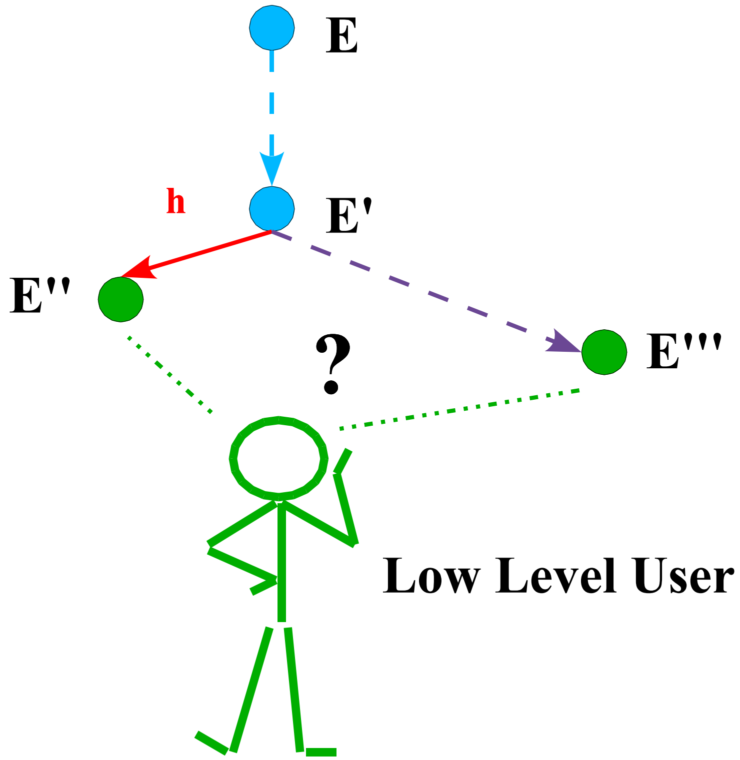
Information Flow and Persistency

- ▷ **Information Flow Security** aims at guaranteeing that no high level (confidential) information is revealed to users at low level, even in the presence of any possible malicious process
- ▷ **Non-Interference**: information does not flow from high to low if the high behavior has no effect on what low level can observe
- ▷ **Dynamicity**: a program which is in a secure state for a certain environment might become unprotected if the environment suddenly changes

Persistency: if a security property is persistent, i.e., a secure process reaches only secure processes, then it ensures security in dynamic contexts

Security as Unwinding - Intuition

If the high level user can perform h reaching E'' from E', then also E''' is reachable from E' and E'' and E''' are undistinguishable for the low level user



Many security properties are instances of this scheme: P_BNDC, SBNDC, CP_BNDC, PP_BNDC, SNDC

Security as Unwinding - Formalization

Let $\sim^l$ be a low level observational equivalence

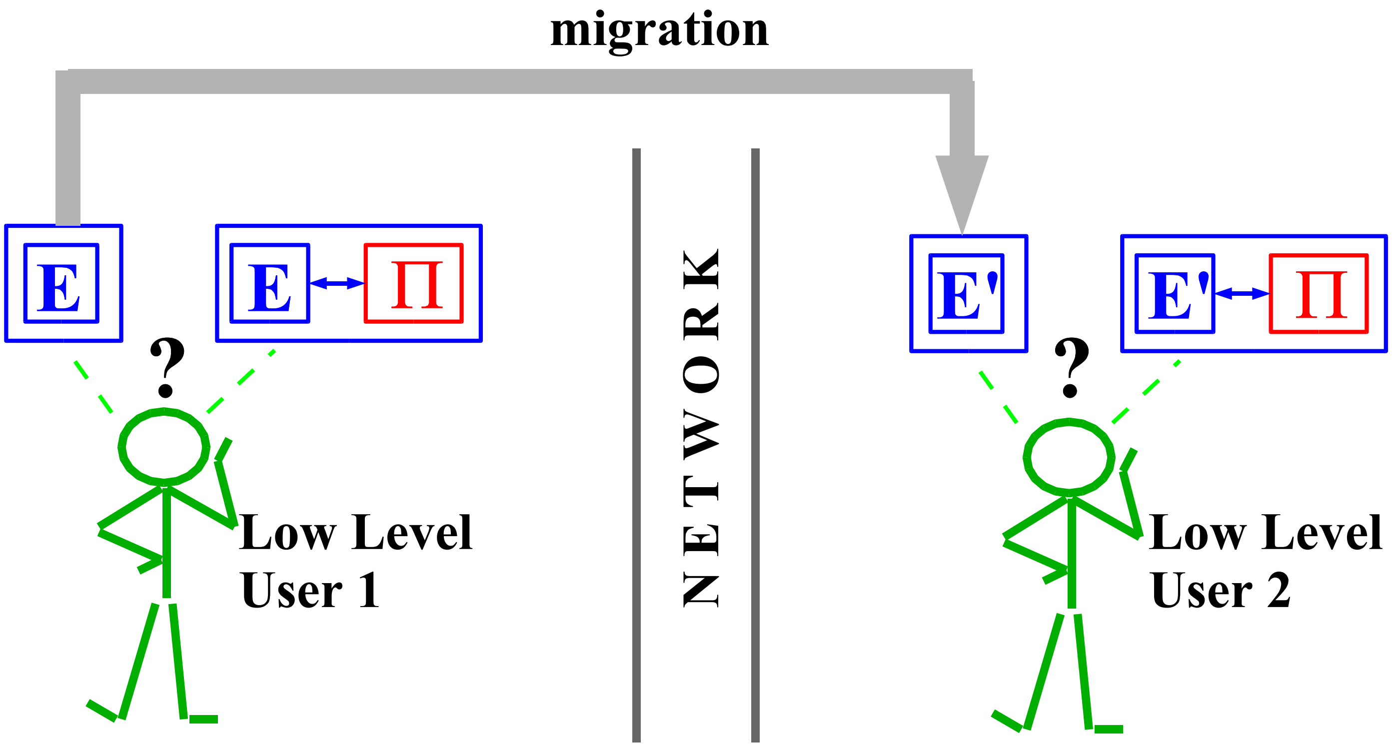
Let $\dashrightarrow$ be a reachability relation

Generalized Unwinding

$$\mathcal{W}(\sim^l, \dashrightarrow) = \{E \in \mathcal{E} \mid \forall F, G \in Reach(E),\ \text{if } F \xrightarrow{h} G \text{ then } \exists G' \text{ such that } F \dashrightarrow G' \text{ and } G \sim^l G'\}$$

The P_BNDC property

Aim: check all the states reachable by the system against all high level (potentially malicious) processes



Persistent BNDC: $\forall\ E'$ reachable from $E, \forall \Pi \in \mathcal{E}_H\ E' \approx^l_B E' | \Pi$

P_BNDC and Unwinding

Weak Bisimulation on Low Actions

$\mathcal{S} \subseteq \mathcal{E} \times \mathcal{E}$ such that if $(E, F) \in \mathcal{S}$ then for all $l \in L \cup \{\tau\}$:

$E \xrightarrow{l} E'$ implies $F \overset{\hat{l}}{\Longrightarrow} F'$ and $(E', F') \in \mathcal{S}$

$F \xrightarrow{l} F'$ implies $E \overset{\hat{l}}{\Longrightarrow} E'$ and $(E', F') \in \mathcal{S}$

$E \approx^{l}_{B} F$ if $(E, F) \in \mathcal{S}$ weak bisimulation on low actions

Silent Reachability

$E \overset{\hat{\tau}}{\Longrightarrow} F$ if E reaches F with a sequence of τ actions.

$$E \in \text{P\_BNDC} \quad \text{if and only if} \quad E \in \mathcal{W}(\approx^{l}_{B}, \overset{\hat{\tau}}{\Longrightarrow})$$

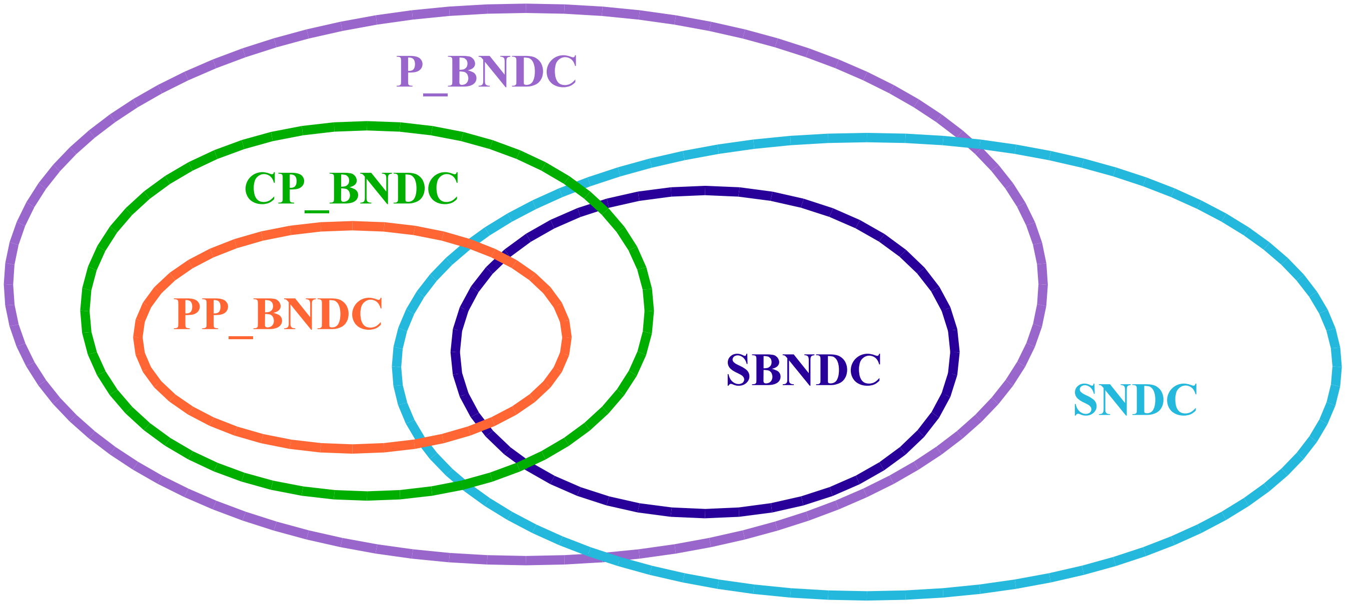
Other Security Properties

SBNDC is equivalent to $\mathcal{W}(\approx^l_B, \equiv)$

CP_BNDC is equivalent to $\mathcal{W}(\approx^l_B, \overset{\tau}{\Longrightarrow})$

PP_BNDC is equivalent to $\mathcal{W}(\approx^l_P, \overset{\tau}{\Longrightarrow})$

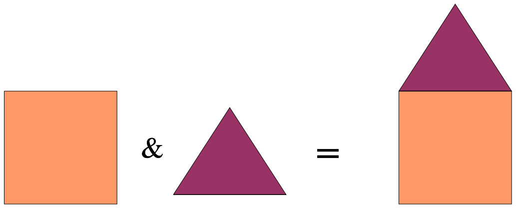
SNDC is equivalent to $\mathcal{W}(\approx^l_T, \equiv)$



Development of Complex Systems

The systematic development of complex systems usually relies on

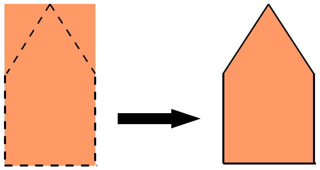
▷ **Composition**: building blocks are put together (e.g., parallel composition)



The composition of secure parts has to be secure as a whole

Compositional Non-Interference properties have been studied

▷ **Refinement**: abstract specifications are refined into more concrete ones



Non-Interference properties based on sets of execution sequences are hard to preserve under refinement

Unwinding and Compositions - General Result

Let f be a partial function and $\odot$ be a relation

f preserves $\odot$ iff

$$G \odot G' \quad \text{implies} \quad (f(G)\uparrow \text{ and } f(G')\uparrow) \quad \text{or} \quad (f(G) \odot f(G'))$$

f reflects $\odot$ iff

$$f(G) \odot M \quad \text{implies} \quad G \odot G' \text{ and } f(G') = M$$

Composition Theorem

If f reflects $\xrightarrow{h}$ and reachability and it preserves $\sim^l$ and $\dashrightarrow$, then $\mathcal{W}(\sim^l, \dashrightarrow)$ is compositional w.r.t. f, i.e.,

$$F \in \mathcal{W}(\sim^l, \dashrightarrow) \quad \text{implies} \quad f(F) \in \mathcal{W}(\sim^l, \dashrightarrow)$$

Unwinding and Compositions - Application

P_BNDC, SBNDC, CP_BNDC, and PP_BNDC

are compositional w.r.t.

$$X \setminus v \qquad X[f] \qquad X|Y$$

The Composition Theorem cannot be applied to $!X$ and $X + Y$

P_BNDC, SBNDC, CP_BNDC, and PP_BNDC are compositional w.r.t. $!X$

CP_BNDC and PP_BNDC are compositional w.r.t. $X + Y$

Horizontal Refinement - Intuition

A refined specification should never show behaviors that were not foreseen in the initial specification

- ▷ each abstract state is refined into at most one concrete state
- ▷ the abstract state simulates its refinement, i.e., if the refinement E of F performs an action a reaching E', then F can perform a reaching F' whose refinement is E'

Horizontal Refinement - Formalization

Simulation

$\mathcal{S} \subseteq \mathcal{E} \times \mathcal{E}$ such that if $(E, F) \in \mathcal{S}$ then for all a:

$E \xrightarrow{a} E'$ implies $F \xrightarrow{a} F'$ and $(E', F') \in \mathcal{S}$

Refinement

$\mathcal{R} \subseteq \mathcal{E} \times \mathcal{E}$ over SPA processes such that:

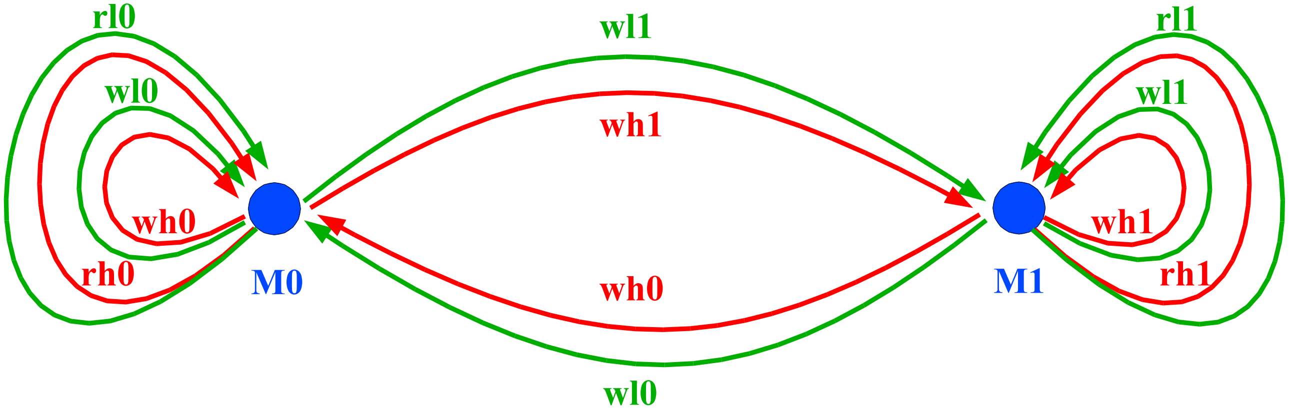
$\mathcal{R}$ is a partial function from $\mathcal{E}$ to $\mathcal{E}$

$\mathcal{R}^{-1}$ is a simulation

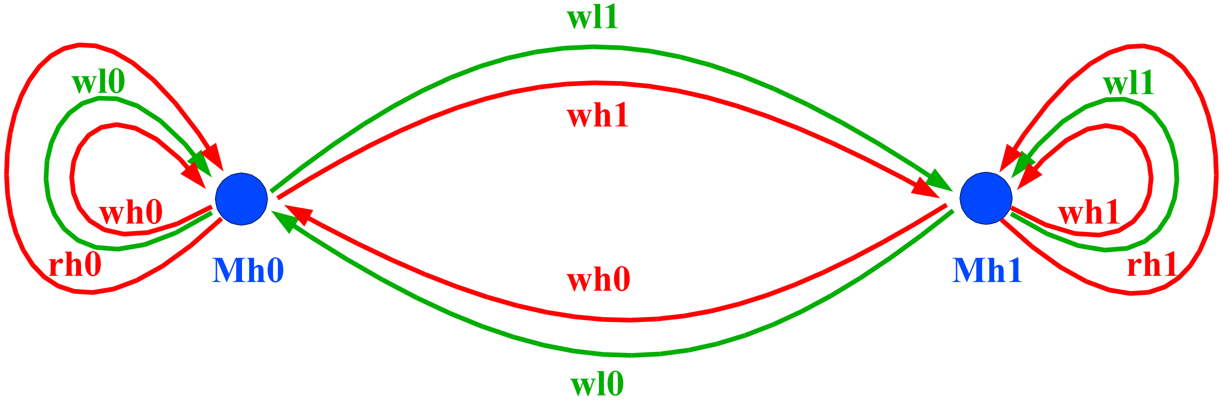
$E \preceq F$, i.e., E is a refinement of F, if there exists a refinement $\mathcal{R}$ such that $\mathcal{R}(F) = E$

Example

Consider a binary memory cell



We refine it into a high level cell by imposing no read up



Properties of the Refinements

- ▷ Composition of Refinements: if $\mathcal{R}_1$ and $\mathcal{R}_2$ are refinements, then $\mathcal{R}_1 \circ \mathcal{R}_2$ is a refinement
- ▷ Refinement and Reachability: if $\mathcal{R}(F) = E$, $\mathcal{R} \cap (Reach(F) \times Reach(E))$ is a refinement
- ▷ Mutual Refinement: if F is finite state and $F \preceq E \preceq F$, $F \sim_B E$
- ▷ Compositionality of Refinement: if $\mathcal{R}(F) = E$ and $\mathcal{R}(G) = I$,
  - ▷ $a.E \preceq a.F$, if $a.F \notin Reach(F)$
  - ▷ $E + I \preceq F + G$, if $F + G \notin Reach(F) \cup Reach(G)$
  - ▷ $E|I \preceq F|G$, $E \setminus v \preceq F \setminus v$, $E[f] \preceq F[f]$

Refinements preserving Unwinding

Unwinding Theorem

Let $\mathcal{R}$ be a refinement preserving $\sim^l$ and $\dashrightarrow$ such that $\mathcal{R}(F) \downarrow$

$$F \in \mathcal{W}(\sim^l, \dashrightarrow) \quad \text{implies} \quad \mathcal{R}(F) \in \mathcal{W}(\sim^l, \dashrightarrow)$$

Composition Theorem

If $\mathcal{R}_1$ and $\mathcal{R}_2$ preserve $\odot$, then $\mathcal{R}_1 \circ \mathcal{R}_2$ preserves $\odot$

Unwinding and Rectification

E not secure $\quad\Rightarrow\quad$ E^s secure

Let s be a sequence of actions such that $E \xrightarrow{s} F$ implies $E \dashrightarrow F$

Given $E = l.F + h.G$ we define

$$E^s = l.F^s + h.G^s + s.G^s$$

Rectification Theorem For all E, $E^s \in \mathcal{W}(\sim^l, \dashrightarrow)$

This can be applied to P_BNDC, CP_BNDC, PP_BNDC with $s = \tau$

Unwinding and Verification

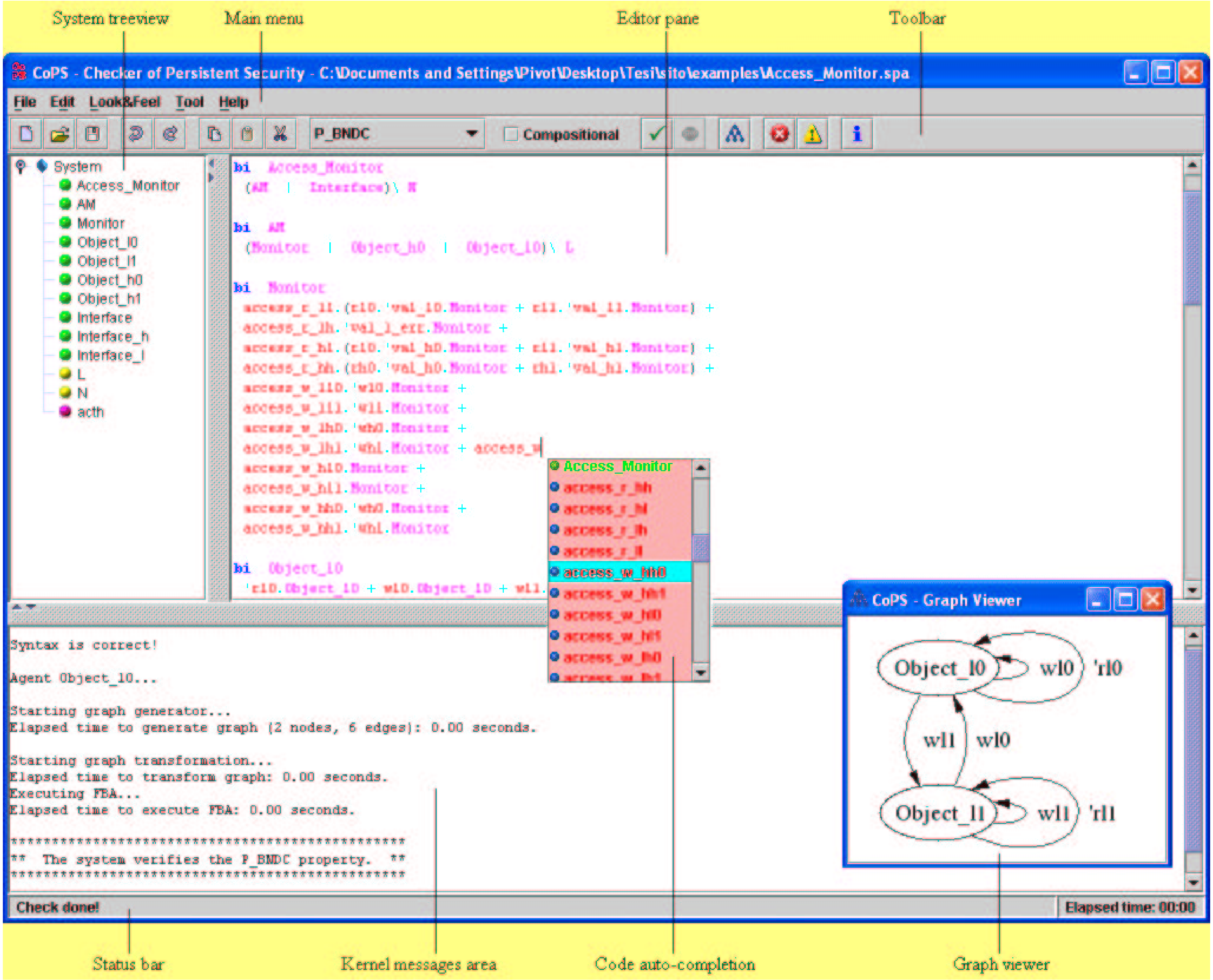
Decidability Theorem

Let E be a finite state process, $\dashrightarrow$ and $\sim^l$ be decidable over finite state processes,

$$E \in W(\sim^l, \dashrightarrow) \text{ is decidable}$$

This is usually inefficient!

To efficiently check P_BNDC, SBNDC, PP_BNDC we use a global bisimulation based characterization implemented in CoPS (see our case-study presentation)



```
bi  Access_Monitor
 (AM  |  Interface)\ N

bi  AM
 (Monitor  |  Object_h0  |  Object_l0)\ L

bi  Monitor
 access_r_ll.(rl0.'val_l0.Monitor + rl1.'val_l1.Monitor) +
 access_r_lh.'val_l_err.Monitor +
 access_r_hl.(rl0.'val_h0.Monitor + rl1.'val_h1.Monitor) +
 access_r_hh.(rh0.'val_h0.Monitor + rh1.'val_h1.Monitor) +
 access_w_ll0.'wl0.Monitor +
 access_w_ll1.'wl1.Monitor +
 access_w_lh0.'wh0.Monitor +
 access_w_lh1.'wh1.Monitor + access_w
 access_w_hl0.Monitor +
 access_w_hl1.Monitor +
 access_w_hh0.'wh0.Monitor +
 access_w_hh1.'wh1.Monitor

bi  Object_l0
 'rl0.Object_l0 + wl0.Object_l0 + wl1.
```

```
Syntax is correct!

Agent Object_l0...

Starting graph generator...
Elapsed time to generate graph (2 nodes, 6 edges): 0.00 seconds.

Starting graph transformation...
Elapsed time to transform graph: 0.00 seconds.
Executing FBA...
Elapsed time to execute FBA: 0.00 seconds.

*************************************************
**  The system verifies the P_BNDC property.  **
*************************************************
```

Check done!

Elapsed time: 00:00

Secure Contexts

$\sim$ observational equivalence, used to equate two processes

$\cdot_l$ low level view which determines

E_l: low level behavior of the process E

$\sim_l$: low level equivalence ($E \sim_l F$ stands for $E_l \sim F_l$)

$\mathcal{C}$ class of contexts, $\mathcal{P}$ class of processes, and X a variable.

$\mathcal{C}$ is secure for $\mathcal{P}$ with respect to X if

$$\forall C[X] \in \mathcal{C}, \forall E \in \mathcal{P}, \quad C[E] \sim_l C[E_l]$$

A low level user cannot discern whether C is interacting with E or E_l

Secure Contexts - II

- ▷ The notion of secure context for a process is parametric, i.e.,
  - ▷ it can be used to restrict the set of possible attackers

    (e.g., if some level passwords cannot be guessed)
  - ▷ it allows to enlarge the set of possible attackers

    (SPA operators can be combined in the contexts construction)
- ▷ We studied two instances: bisimulation and trace equivalence
- ▷ We showed that BNDC and NDC are instances of our notion

Conclusions

- ▷ we considered Unwinding conditions defining security properties
- ▷ we analyzed how to
  - ▷ incrementally build secure systems via
    - ∗ composition
    - ∗ refinement
  - ▷ rectify unsecure systems
  - ▷ efficiently verify security
- ▷ we implemented a tool for efficient security verification
- ▷ we considered Secure Contexts to relax the security conditions

References 2002

- ▷ R. Focardi and S. Rossi. Information Flow Security in Dynamic Contexts CSFW 2002, IEEE, pagg. 307–319.
- ▷ R. Focardi, C. Piazza, and S. Rossi. Proofs Methods for Bisimulation based Information Flow Security VMCAI 2002, LNCS 2294, pagg. 16–31.
- ▷ A. Bossi, R. Focardi, C. Piazza, and S. Rossi. Transforming Processes to Check and Ensure Information Flow Security AMAST 2002, LNCS 2422 , pagg. 271–286.
- ▷ A. Bossi, R. Focardi, C. Piazza, and S. Rossi. A Proof System for Information Flow Security LOPSTR 2002, LNCS 2264, pagg. 199–218.

References 2003

- ▷ A. Bossi, R. Focardi, C. Piazza, and S. Rossi. Bisimulation and Unwinding for Verifying Possibilistic Security Properties VMCAI 2003, LNCS 2575, pagg. 223–237.
- ▷ A. Bossi, D. Macedonio, C. Piazza, and S. Rossi. Information Flow Security and Recursive Systems ICTCS 2003, LNCS ??, pagg. ??.
- ▷ A. Bossi, R. Focardi, C. Piazza, and S. Rossi. Refinement Operators and Information Flow Security SEFM 2003, IEEE, pagg. 44–53.
- ▷ A. Bossi, D. Macedonio, C. Piazza, and S. Rossi. Secure Contexts for Confidential Data CSFW 2003, IEEE, pagg. 14–25.
- ▷ A. Bossi, R. Focardi, C. Piazza, and S. Rossi. Verifying Persistent Security Properties To appear in Computer Languages, Systems and Structures
- ▷ C. Piazza, E. Pivato, and S. Rossi. CoPS - Checker of Persistent Security Submitted to conference.