

Decentralized control for secrecy

joint work with

E. Badouel B. Caillaud
Inria-Rennes

M. Bednarczyk A. Borzyszkowski
Ipan-Gdansk

February 2007

Works in the field of Security

- Many aspects (access control, encryption, authentication, trust, intrusion detection, leaks of information ...)
- Static Verification (Cryptographic Protocols) with rewriting techniques, model checking, information theory...
- Run time Verification (User Requests in Web Services) using type and effect systems, automata...
- Any contribution of SCT?

This presentation

- A first attempt in this direction
- Partial results, toy applications

Thesis

Supervisory Control Theory may help enforcing Security

Supervisory Control

Usual Game

Control Objective : Safety / Liveness

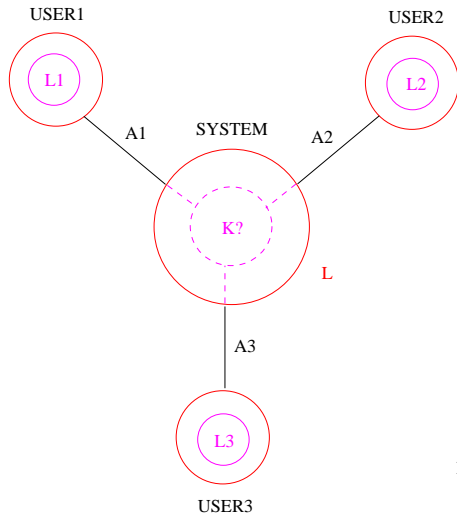
Observers: on the side of the Controller

Other Game

Control Objective : Secrecy

Observers: on the side of the Opponent

Example 1



SECRETS

$L_i \text{ IN } A_i^*$

UNCONTROLLED BEHAVIOUR

$L \text{ INCLUDED IN } (A_1+A_2+A_3)^*$

FIND MAXIMAL PERMISSIVE CONTROL

K INCLUDED IN L SUCH THAT

USERS $i+1$ AND $i+2$ MAY NEVER KNOW

THAT USER i HAS PERFORMED w_i IN L_i

EVEN THOUGH THEY TALK TO EACH OTHER

Formalization

SECRET SET

$$S_1 = L_1 \parallel (A_2 + A_3)^* \cap L$$

$$S_2 = L_2 \parallel (A_1 + A_3)^* \cap L$$

$$S_3 = L_3 \parallel (A_1 + A_2)^* \cap L$$

OPPONENT'S ALPHABET

$$\Sigma_1 = A_2 \cup A_3$$

$$\Sigma_2 = A_1 \cup A_3$$

$$\Sigma_3 = A_1 \cup A_2$$

$\mathcal{S} = \{(S_1, \Sigma_1), (S_2, \Sigma_2), (S_3, \Sigma_3)\}$ is a **CONCURRENT SECRET**

Definition

\mathcal{S} is **opaque** if $\forall w \in L \forall i$

$w \in S_i \Rightarrow \Pi_{\Sigma_i}(w) = \Pi_{\Sigma_i}(w')$ for some $w' \in L \setminus S_i$

opacity is the opposite of normality when $(\forall i) S_i = \overline{S_i}$

S_i is **normal** if $\Pi_{\Sigma_i}(w) = \Pi_{\Sigma_i}(w') \Rightarrow w \in \overline{S_i}$ iff $w' \in \overline{S_i}$

An earlier definition of opacity

Bryans, Koutny, Mazare, Ryan

Definition

A predicate ϕ over runs ρ of the system is opaque w.r.t. the observation function obs if, for every run $\rho \in \phi$, there is a run $\rho' \notin \phi$ such that $obs(\rho) = obs(\rho')$

single observer

arbitrary observation function (states may be observable)

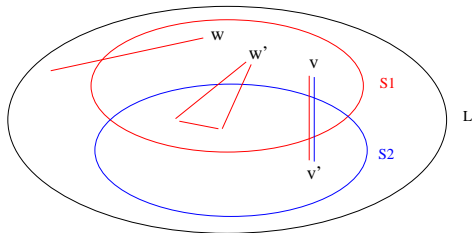
opacity is in general not decidable

opacity is asymmetric

Concurrent opacity is needed in order that the observer
neither knows ϕ , nor he knows **not** ϕ

Safe Kernels

If L is **prefix closed** and all secrets S_i are **regular**, one can **decide** whether the concurrent secret S is **opaque** w.r.t. L .
If not, one can **compute the safe kernel** $K(L, S)$ of L .



w in $K(L, S)$

w' not in $K(L, S)$

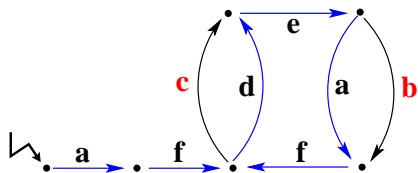
v, v' in $K(L, S)$

Definition

The *safe kernel* $K(L, S)$ of L is the subset of all words $w \in L$ such that for every prefix u of w and for every i
 $\Pi_{\Sigma_i}(u) = \Pi_{\Sigma_i}(u')$ for some $u' \in L \setminus S_i$

Example 2

But using $K(L, S)$ as a controller does not solve our problem ... because users know the system and the controller!



$S_1 = \Sigma^* afc(\Sigma \setminus \{c\})^*$ (last c follows af), $\Sigma_1 = \{c, f\}$,

$S_2 = \Sigma^* deb(\Sigma \setminus \{b\})^*$ (last b follows de), $\Sigma_2 = \{b, e\}$

$K(L, S) = L \setminus afc\Sigma^*$

$K(K(L, S), S) = K(L, S) \setminus afdeb\Sigma^*$

What remains in the end is $(afde)^*$

Supremal Safe Sublanguage

$SupK(\bullet, \mathcal{S})$ is monotone in first argument

Definition

Let $SupK(L, \mathcal{S})$ be the greatest fixpoint of the operator $K(\bullet, \mathcal{S})$ included in L

Theorem

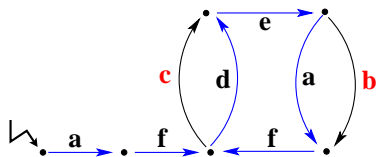
$SupK(L, \mathcal{S})$ is the union of all controls enforcing the opacity of concurrent secret \mathcal{S}

Sufficient conditions under which $SupK(L, \mathcal{S})$ is regular and computable ?

Two sources of problems

- The closure ordinal of $K(\bullet, \mathcal{S})$ may be transfinite
- $SupK(\bullet, \mathcal{S})$ may be not regular

$K(\bullet, \mathcal{S})$ has a transfinite closure ordinal



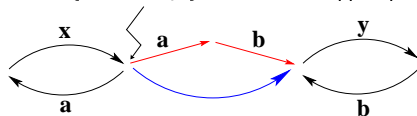
$S_1 = \Sigma^*afc(\Sigma \setminus \{c\})^*$ (last c follows af), $\Sigma_1 = \{c, f\}$,
 $S_2 = \Sigma^*deb(\Sigma \setminus \{b\})^*$ (last b follows de), $\Sigma_2 = \{b, e\}$
 $S_3 = L \setminus (\Sigma^*c\Sigma^*)$ (there is no c)
 S_3 safe w.r.t. any $L' \subseteq L$ with at least one word with c

$$\lim_{i \rightarrow \omega} K^i(L, \mathcal{S}) = \text{Pref}((afde)^\omega)$$

$$K^{\omega+1}(L, \mathcal{S}) = \emptyset$$

$SupK(\bullet, \mathcal{S})$ is not regular

$$\Sigma = \{a, b, x, y\} \quad L = Pref((ax)^*(\varepsilon + ab)(yb)^*)$$



$$\Sigma_1 = \{a, b\}, \quad \mathcal{CS}_1 = \varepsilon + (ax)^* ab(yb)^* + \{a, x, y\}^*$$

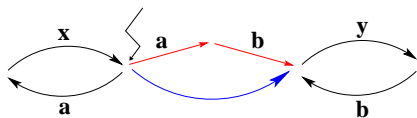
$$\Sigma_2 = \{x, y\}, \quad \mathcal{CS}_2 = (ax)^*(yb)^*$$

$$\Sigma_3 = \{a, b, x, y\}, \quad \mathcal{CS}_3 = \varepsilon + a\Sigma^*$$

$$S_1 = \rightarrow$$

$$S_2 = \rightarrow \rightarrow$$

$$SupK(L, \mathcal{S}) = Pref(\bigcup_{n \in \mathbb{N}} (ax)^n (\varepsilon + ab)(yb)^n)$$



$$(ax)^n (\varepsilon) (yb)^m$$

$$\downarrow \{a,b\}$$

$$(ax)^{n-1} (ab) (yb)^{m-1}$$

$$\downarrow \{x,y\}$$

$$(ax)^{n-1} (\varepsilon) (yb)^{m-1}$$

$$\downarrow \{a,b\}$$

$$(ax)^{n-2} (ab) (yb)^{m-2}$$

...

hence $n = m$

Some sufficient conditions

language theoretic conditions (i) and (ii)

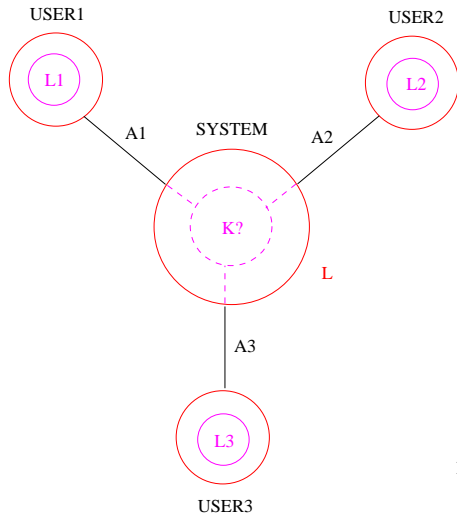
- i) system language L closed under prefix
- ii) **secrets closed under suffix** ($S_i \Sigma^* \subseteq S_i$)

structural conditions (iii) or (iv) or (v)

- iii) $\Sigma_1 \subseteq \Sigma_2 \dots \subseteq \Sigma_n$ chain of alphabets
- iv) $S_1 \subseteq S_2 \dots \subseteq S_n$ chain of secrets
- v) $(\forall i \neq j) (\forall w, w' \in L)$ observers \perp secrets
- $\Pi_{\Sigma_j}(w) = \Pi_{\Sigma_j}(w') \Rightarrow w \in S_i \text{ iff } w' \in S_i$ true in Example 1

do not hold for Example 2!

Example 1



SECRETS

$L_i \text{ IN } A_i^*$

UNCONTROLLED BEHAVIOUR

$L \text{ INCLUDED IN } (A_1+A_2+A_3)^*$

FIND MAXIMAL PERMISSIVE CONTROL

K INCLUDED IN L SUCH THAT

USERS $i+1$ AND $i+2$ MAY NEVER KNOW

THAT USER i HAS PERFORMED w_i IN L_i

EVEN THOUGH THEY TALK TO EACH OTHER

Orthogonality in Example 1

SECRET SET

$$S_1 = L_1 \parallel (A_2 + A_3)^* \cap L$$

$$S_2 = L_2 \parallel (A_1 + A_3)^* \cap L$$

$$S_3 = L_3 \parallel (A_1 + A_2)^* \cap L$$

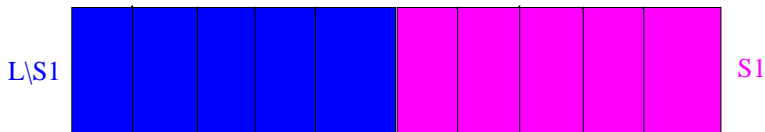
OPPONENT'S ALPHABET

$$\Sigma_1 = A_2 \cup A_3$$

$$\Sigma_2 = A_1 \cup A_3$$

$$\Sigma_3 = A_1 \cup A_2$$

$$L_1 \subseteq A_1^* \quad L_2 \subseteq A_2^* \quad L_3 \subseteq A_3^*$$

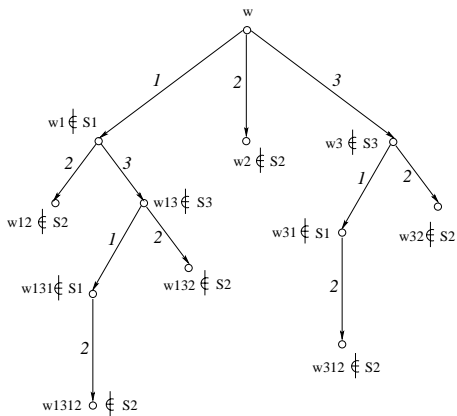


Equivalence Classes w.r.t. Observer 3

$$\Pi_{\Sigma_3}(u) = \Pi_{\Sigma_3}(u') \Rightarrow \Pi_{A_1}(u) = \Pi_{A_1}(u')$$

hence $u \in S_1$ if and only if $u' \in S_1$

$S_1 \subseteq S_2$ $\Sigma_3 \subseteq \Sigma_2$ $Obs_1 \perp S_3$ (a mixed case)

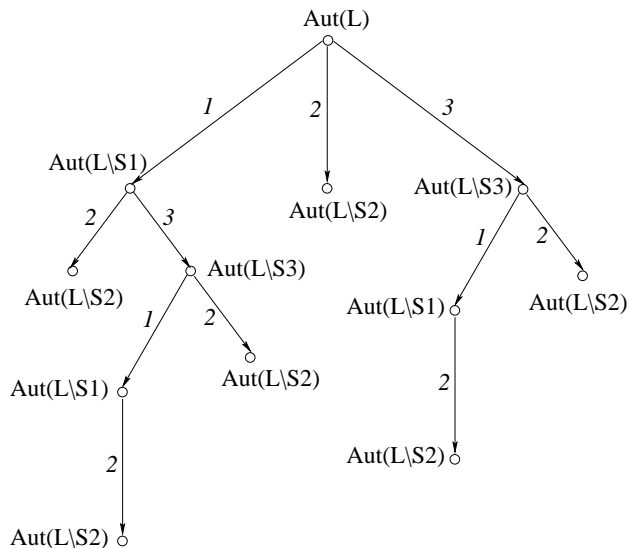


Finite pattern of proofs for $w \in SupK(L, S)$

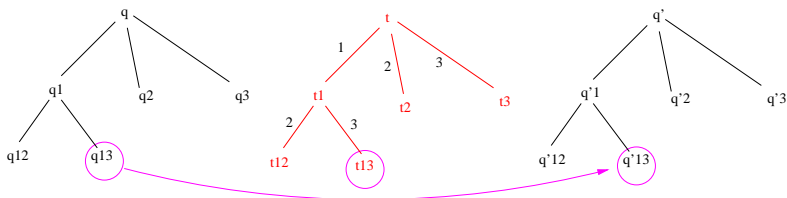
Theorem

If there exists a finite number of patterns of proof for all $w \in \text{SupK}(L, \mathcal{S})$, then $\text{SupK}(L, \mathcal{S})$ is a regular language

Constructing an automaton from a pattern



Synchronized moves



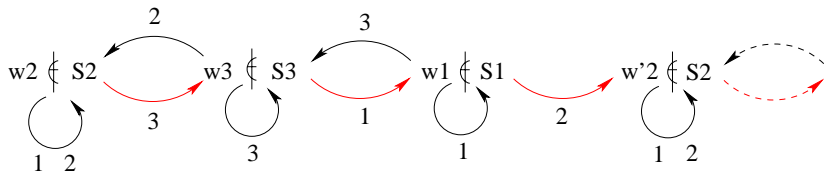
$t_{ij} \in (\Sigma_k)$ or $t_{ij} = \varepsilon$

$t_{ij} \in (\Sigma_k)$ and $t_{ijk} \in (\Sigma_k) \Rightarrow tt_{ij} = t_{ijk}$

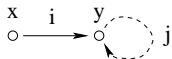
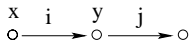
Compute the projection on topmost nodes

A case where finite patterns are not enough

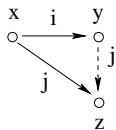
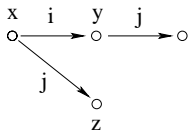
$$S_1 \subseteq S_2 \quad \Sigma_2 \subseteq \Sigma_3 \quad Obs_1 \perp S_3$$



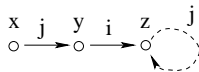
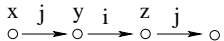
The four rules



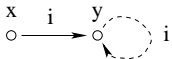
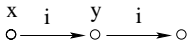
Secret j included in Secret i



Sigma j included in Sigma i



Observator i ortho Secret j



True

Theorem

If the complete n -ary tree rewrites to some finite graph, the spanning tree of this graph is a uniform pattern of proofs for all $w \in \text{SupK}(L, \mathcal{S})$

Theorem

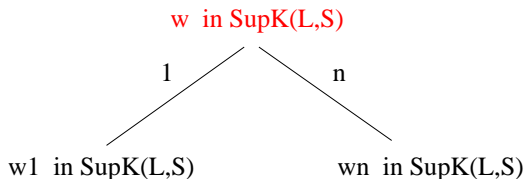
It is decidable whether some finite graph may be derived from the complete n -ary tree, and such graphs may be computed when they exist

One can then construct a finite automaton accepting $\text{SupK}(L, \mathcal{S})$

Decentralized control

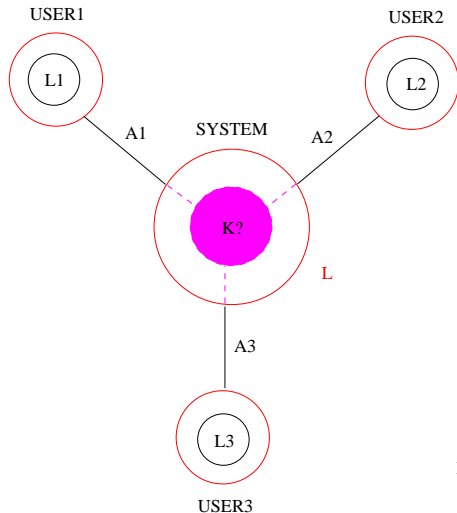
Theorem

Let $w \in L$. If, for all $i \in \{1, \dots, n\}$, $\pi_i(w) = \pi_i(w_i)$ for some $w_i \in \text{SupK}(L, S)$ then $w \in \text{SupK}(L, S)$



BY DEFINITION OF
THE
GREATEST FIXED POINT

Example 1



SECRETS
 $L_i \text{ IN } A_i^*$

UNCONTROLLED BEHAVIOUR

$L \text{ INCLUDED IN } (A_1+A_2+A_3)^*$

FIND MAXIMAL PERMISSIVE CONTROL

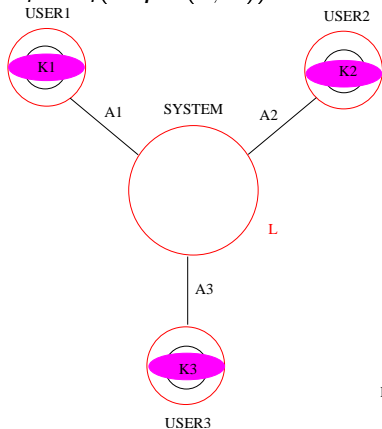
K INCLUDED IN L SUCH THAT

USERS $i+1$ AND $i+2$ MAY NEVER KNOW
THAT USER i HAS PERFORMED w_i IN L_i

EVEN THOUGH THEY TALK TO EACH OTHER

Example 1

$$K_i = \pi_i(\text{SupK}(L, S))$$



SECRETS

$L_i \text{ IN } A_i^*$

UNCONTROLLED BEHAVIOUR

$L \text{ INCLUDED IN } (A_1 + A_2 + A_3)^*$

MAXIMAL PERMISSIVE CONTROLS

$K_i \text{ INCLUDED IN } A_i^* \text{ SUCH THAT}$

USERS $i+1$ AND $i+2$ MAY NEVER KNOW

THAT USER i HAS PERFORMED w_i IN L_i

EVEN THOUGH THEY TALK TO EACH OTHER