Security issues in DES: non-interference and opacity

From observability to privacy and security in discrete event systems

Prof. Gianmaria De Tommasi
Email: detommas@unina.it

December 2020

DIETI. UNIVERSITA' DEGLI STUDI DI NAPOLI FEDERICO II
DIPARTIMENTO DI INGEGNERIA ELETTRICA
E DELLE TECNOLOGIE DELL'INFORMAZIONE

Course syllabus

1. Discrete Event Systems (DES), Languages and Automata
2. Petri nets (PNs) and their twofold representation to model DES
3. MILP and ILP formulations: logical conditions, binary variables “do everything", and variable connecting
4. Adding uncertainty: unobservable events and observers for finite state automata and PNs
5. Augmenting the observers: diagnosability of prefix-closed languages, diagnosers and the fault detection for finite state automata
6. Diagnosability and fault detection in PNs - Part I: graph-based approaches
7. Diagnosability and fault detection in PNs - Part II: algebraic approaches for bounded systems
8. **Security issues in DES: non-interference and opacity**
9. Non-interference and opacity enforcement
10. Open issues

Outline

1. The non-interference property
   - The automata case
   - The Petri nets case
2. Opacity
   - The automata case
   - The Petri net case
3. Example

Security issues in DES

- In system security it is important **to prevent information leaks**
- Various *information flow properties* have been defined in the literature
  - anonymity
  - non-interference
  - secrecy
  - privacy
  - security
  - opacity
- **Objective:** to prevent to an **intruder** to access to *secret*
- DES have been used to model different of the above properties
  - non-interference (*separation* among different domains)
  - opacity (the secret is a set of specific states or sequences)

Verification vs enforcement of a security property

Verification problem is the analysis problem → given a model of the process assess a given property

Enforcement problem is the synthesis problem → if a process does not satisfy a given property and it is not possible to re-design the process itself, then

- supervisory control approaches can be used to limit the behaviour of the system in closed-loop. . .
- . . .as well as obfuscation or insertion techniques. . .
- . . .we will see enforcement during the next lecture

Non-interference

- Two classes of users: **high-level** and **low-level** users
- It is assumed that both high-level and low-level users know the system structure (model) but they interact with the system in two different ways (*views*)
- A leak of information occurs when a low-level user (the **intruder**) obtains information meant to be visible only to high-level ones
- If the high-level view of the system *interferes* with the low-level one, information leaks may occur
- The high-level and low-level events corresponds to different domains
- The non-interference framework can be extended to the **multilevel** (multi domain) case

Low- and high-level events

- The two classes of users induces a partition on the event set E
- E_L is the set of $low-level$ events, the only ones that can be observed by the **low-level users**
- E_H is the set of $high-level$ events, the ones that cannot be observed by the **low-level users**
- The high-level users can observe **all** the events
- $E = E_L \cup E_H$ and $E_L \cap E_H = \emptyset$

Toward a language-based definition of non-interference

- If the process/plant is modelled as the automaton

$$G = (X\,,\, E_L \cup E_H\,,\, f\,,\, x_0)$$

where $\mathcal{L}(G)$ denotes the generated language

- By replacing all the high-level events with the empty string ε, a **nondeterministic** automaton can be derived

$$G_{nd}^{E_L} = \left(X\,,\, E_L \cup \{\varepsilon\}\,,\, f_{nd}^{E_L}\,,\, x_0\right)$$

with $\mathcal{L}\left(G_{nd}^{E_L}\right) = Pr_L\left(\mathcal{L}(G)\right) \subseteq E_L^*$, where $Pr_L(\cdot)$ is the usual projection function on the set of low-level events E_L

  - $Pr_L(\varepsilon) = \varepsilon$
  - $Pr_L(\sigma e) = \{ \quad Pr_L(\sigma)e \quad \text{if } e \in E_L \; \mathrm{Pr}_L(\sigma)$ otherwise

- By removing the high-level events a **deterministic** automaton can be derived

$$G^{E_L} = \left(X\,,\, E_L\,,\, f^{E_L}\,,\, x_0\right)$$

where

  - $\mathcal{L}\left(G^{E_L}\right) \subseteq E_L^*$
  - **and $\mathcal{L}\left(G^{E_L}\right) \subseteq \mathcal{L}\left(G_{nd}^{E_L}\right)$, by definition**

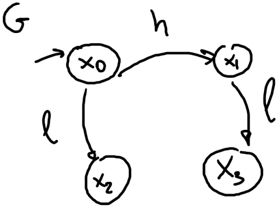
Strong non-deterministic non-interference (SNNI)

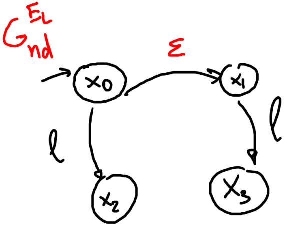
- An automaton G is **strong non-deterministic non interferent (SNNI)** *if and only if*

$$\mathcal{L}\left(G_{nd}^{E_L}\right) \subseteq \mathcal{L}\left(G^{E_L}\right)$$

- equivalent definitions
  - G SNNI $\Leftrightarrow \mathcal{L}\left(G_{nd}^{E_L}\right) = \mathcal{L}\left(G^{E_L}\right)$
  - G SNNI $\Leftrightarrow Pr_L\left(\mathcal{L}(G)\right) = \mathcal{L}\left(G^{E_L}\right)$

Example - A SNNI automaton

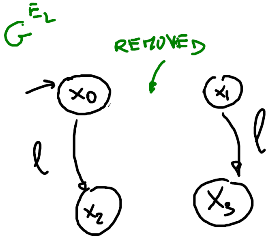




$E_L = \{\ell\}$; $E_H = \{h\}$

$\mathcal{L}(G) = \{\varepsilon, \ell, h, h\ell\}$

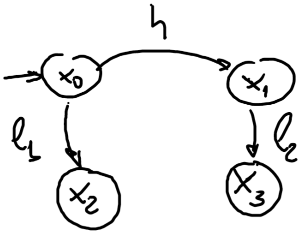
$\underline{\mathcal{L}(G_{nd}^{E_L}) = \{\varepsilon, \ell\}} = P_L(\mathcal{L}(G))$



$\underline{\mathcal{L}(G^{E_L}) = \{\varepsilon, \ell\}}$

G IS SNNI

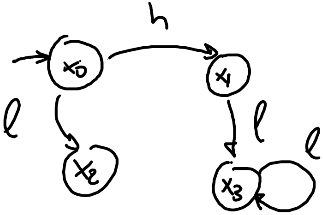
Example - Not SNNI automata



$E_L = \{l_1, l_2\}$

$E_H = \{h\}$

$\mathcal{L}(G) = \{\varepsilon, l_1, h; hl_2\}$



$E_L = \{l\}$

$E_H = \{h\}$

$\mathcal{L}(G) = \{\varepsilon, l, h, hl; hll; \dots\}$

An example from literature

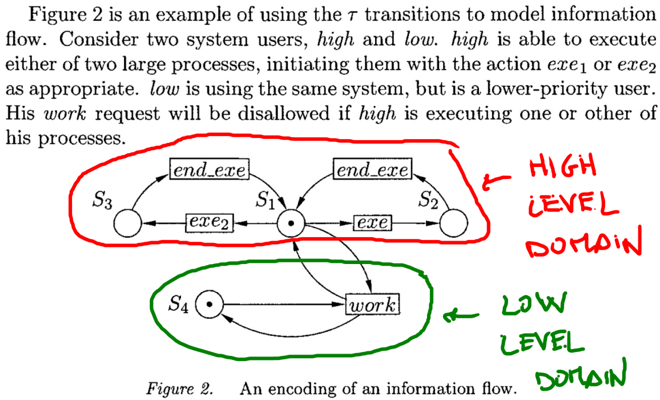


Figure 2 is an example of using the τ transitions to model information flow. Consider two system users, *high* and *low*. *high* is able to execute either of two large processes, initiating them with the action exe_1 or exe_2 as appropriate. *low* is using the same system, but is a lower-priority user. His *work* request will be disallowed if *high* is executing one or other of his processes.

Figure 2. An encoding of an information flow.

Example taken from

J. W. Bryans, M. Koutny and P. Y. A. Ryan,

Modelling dynamic opacity using Petri nets with silent actions,

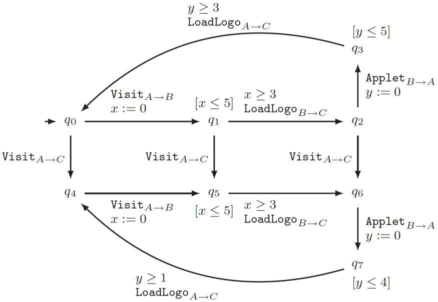
IFIP World Computer Congress, 2004

A timed example

- The attacks allow a malicious website to determine whether or not the user has recently visited some other unrelated webpage. Example: an insurance company site could determine whether the user has recently visited websites relating to a particular medical condition
- Alice is surfing on the web, and visits Bob's website
- Bob wants to find out whether Alice has visited Charlie's website
- First, Bob looks at Charlie's site, and picks a file that any visitor to the site will have seen (for example the file `logo.jpg`)
- Bob is going to determine whether the logo file is in Alice's browser cache, since if the file is in her cache, then she must have visited Charlie's website recently
- Bob writes a Java applet that implements his attack, and embeds it in his home page
- The applet measures the time required to access `logo.jpg` on Alice's machine, and reports this time back to Bob
- According to this time, Bob may conclude that Alice has been to Charlie's site recently

Example taken from

G. Benattar, F. Cassez, D. Lime and O. H. Roux
Control and synthesis of non-interferent timed systems,
International Journal of Control, 2015


$y \geq 3$
$\mathtt{LoadLogo}_{A\to C}$
$[y \leq 5]$
q_3
$\mathtt{Applet}_{B\to A}$
$y := 0$
$\mathtt{Visit}_{A\to B}$
$x := 0$
$[x \leq 5]$
$x \geq 3$
$\mathtt{LoadLogo}_{B\to C}$
q_0
q_1
q_2
$\mathtt{Visit}_{A\to C}$
$\mathtt{Visit}_{A\to C}$
$\mathtt{Visit}_{A\to C}$
q_4
q_5
q_6
$\mathtt{Visit}_{A\to B}$
$x := 0$
$[x \leq 5]$
$x \geq 3$
$\mathtt{LoadLogo}_{B\to C}$
$\mathtt{Applet}_{B\to A}$
$y := 0$
q_7
$[y \leq 4]$
$y \geq 1$
$\mathtt{LoadLogo}_{A\to C}$

Assess non-interference in DES modeled as automata

- SNNI implies that

$$Pr_L\left(\mathcal{L}(G)\right) = \mathcal{L}\left(G^{E_L}\right) \tag{1}$$

- For untimed automata condition (1) can be checked in polynomial time by means of operation between the automata G and G^{E_L}
- It has been proved that SNNI assessment in timed-automata is EXPTIME-complete

G. Benattar, F. Cassez, D. Lime and O. H. Roux
Control and synthesis of non-interferent timed systems,
International Journal of Control, 2015

Recast SNNI definition to the case of P/T nets

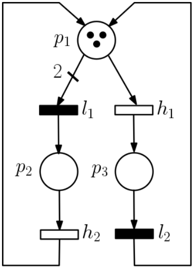
Main assumptions

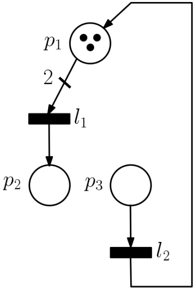
- The net system $\mathcal{S} = \langle N, \boldsymbol{m}_0 \rangle$ is bounded
- The net system is assumed to be unlabeled
- The P/T net: $N = (P, L, H, \mathbf{Pre}, \mathbf{Post})$, with $L \cap H = \emptyset$
    - L low-level transitions
    - H high-level transitions
    - $T = L \cup H$

Objective: to exploit the twofold representation of PN systems to find algebraic conditions to assess SNNI

Low-level system

- The *low-level* system is the system *induced* by the low-level transitions
- $L = \{l_1, l_2\}$ and $H = \{h_1, h_2\}$





SNNI net system

- Let $\mathcal{S} = \langle N , \boldsymbol{m}_0 \rangle$ be a net system and $\mathcal{S}_L = \langle N_L , \boldsymbol{m}_0 \rangle$ the correspondent low-level system
- $\mathcal{S}$ is SNNI *if and only if*

$$Pr_L\left(\mathcal{L}(N , \boldsymbol{m}_0)\right) = \mathcal{L}(N_L , \boldsymbol{m}_0)$$

Exploiting the boundedness assumption

- The system is assumed **bounded**
- Therefore it is possible to describe the state space by means of a **set of linear constraints** (as we did for diagnosability)
- There exists a set of ρ integer vectors $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_\rho$ with $\rho \leq |\sigma|$ such that the following linear constraints are fulfilled

$$\begin{cases} \boldsymbol{m} \geq \mathbf{Pre} \cdot \boldsymbol{s}_1 \\ \boldsymbol{m} + \boldsymbol{C} \cdot \boldsymbol{s}_1 \geq \mathbf{Pre} \cdot \boldsymbol{s}_2 \\ \ldots \\ \boldsymbol{m} + \boldsymbol{C} \cdot \sum_{i=1}^{\rho-1} \boldsymbol{s}_i \geq \mathbf{Pre} \cdot \boldsymbol{s}_\rho \\ \sum_{i=1}^{\rho} \boldsymbol{s}_i = \pi(\sigma) \end{cases} \quad (2)$$

  iff there exists at least one sequence σ, which is enabled under the marking $\boldsymbol{m}$ and such that $\pi(\sigma) = \boldsymbol{\sigma}$
- **For a bounded net, given a sufficiently large number of inequality constraints** (2)**, it is possible to describe the** $R(N, \boldsymbol{m}_0)$ **set** → **let assume that** J **inequalities are sufficient to this purpose**

The key idea exploited in the algebraic approach

- For bounded net, given J there exists a maximum number of time a transition can fire given the constraints (2)
- Let us denote as φ_t the maximum number of firings of a **low-level transition** t in the low-level system $\mathcal{S}_L$
- If it is possible to have at least one additional firing of t in the original net system, this implies interference
- The other source of interference is the possibility of using high-level transitions to enable the firing of t

Maximum number of firing of a low-level transition in $\mathcal{S}_L$

Given J constraints in (2), the maximum number of firings for $t \in L$ in $\mathcal{S}_L$ can be computed as the solution of the ILP

$$\varphi_t = \max \sum_{i=1}^{J} \boldsymbol{\sigma}_i(t)$$

subject to

$$\begin{cases} \boldsymbol{m}_0 \geq \mathbf{Pre}_L \cdot \boldsymbol{\sigma}_1 \\ \boldsymbol{m}_0 + \boldsymbol{C}_L \cdot \boldsymbol{\sigma}_1 \geq \mathbf{Pre}_L \cdot \boldsymbol{\sigma}_2 \\ \dots \\ \boldsymbol{m}_0 + \boldsymbol{C}_L \cdot \sum_{i=1}^{J-1} \boldsymbol{\sigma}_i \geq \mathbf{Pre}_L \cdot \boldsymbol{\sigma}_J \\ \boldsymbol{m}_0 + \boldsymbol{C}_L \cdot \sum_{i=1}^{J} \boldsymbol{\sigma}_i \geq \mathbf{0} \\ \boldsymbol{\sigma}_i \in \mathbb{N}^n , \quad i = 1\,, 2\,, \dots\,, J \end{cases}$$

SNNI assessment in DES modeled as Petri nets (I)

Given a K-bounded system $\mathcal{S}$, let consider the two ILP problems

$$\min \sum_{i=1}^{J} \sum_{t_h \in H} \boldsymbol{x}_i(t_h) \tag{3}$$

subject to

$$\mathcal{X}\left(\boldsymbol{m}_0\,,\varphi_t\right):\begin{cases} \boldsymbol{m}_0 \geq \textbf{Pre}\cdot\boldsymbol{x}_1 \\ \boldsymbol{m}_0 + \boldsymbol{C}\cdot\boldsymbol{x}_1 \geq \textbf{Pre}\cdot\boldsymbol{x}_2 \\ \dots & \text{(4a)} \\ \boldsymbol{m}_0 + \boldsymbol{C}\cdot\sum_{i=1}^{J-1}\boldsymbol{x}_i \geq \textbf{Pre}\cdot\boldsymbol{x}_J \\ \boldsymbol{m}_0 + \boldsymbol{C}\cdot\sum_{i=1}^{J}\boldsymbol{x}_i \geq \boldsymbol{0} \\ \sum_{i=1}^{J}\boldsymbol{x}_i(t) \geq \varphi_t + 1 & \text{(4b)} \\ \boldsymbol{x}_i \in \mathbb{N}^n\,, \quad i = 1\,,2\,,\,\dots\,,J & \text{(4c)} \end{cases}$$

$$\min\left[\sum_{i=1}^{J}\sum_{t_l\in L}\boldsymbol{y}_i(t_l) + \epsilon\sum_{i=1}^{J}\sum_{t_h\in H}\boldsymbol{y}_i(t_h)\right] \tag{5}$$

subject to

$$\mathcal{Y}\left(\boldsymbol{m}_0\,,\varphi_t\right):\begin{cases} \boldsymbol{m}_0 \geq \textbf{Pre}\cdot\boldsymbol{y}_1 \\ \boldsymbol{m}_0 + \boldsymbol{C}\cdot\boldsymbol{y}_1 \geq \textbf{Pre}\cdot\boldsymbol{y}_2 \\ \dots & \text{(6a)} \\ \boldsymbol{m}_0 + \boldsymbol{C}\cdot\sum_{i=1}^{J-1}\boldsymbol{y}_i \geq \textbf{Pre}\cdot\boldsymbol{y}_J \\ \boldsymbol{m}_0 + \boldsymbol{C}\cdot\sum_{i=1}^{J}\boldsymbol{y}_i \geq \boldsymbol{0} \\ \sum_{i=1}^{J}\boldsymbol{y}_i(t) = \varphi_t & \text{(6b)} \\ \boldsymbol{y}_i \in \mathbb{N}^n\,, \quad i = 1\,,2\,,\,\dots\,,J & \text{(6c)} \end{cases}$$

with $\epsilon < (K\cdot\text{card}\,(H)\cdot J)^{-1}$

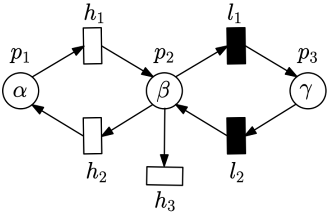
SNNI assessment in DES modeled as Petri nets (II)

System $\mathcal{S}$ is SNNI **iff** the following two conditions hold for each $t \in L$

1) the ILP problem (3)-(4) does not admit a solution

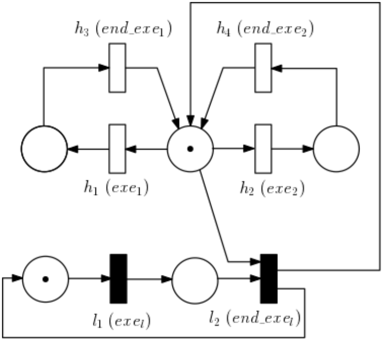
2) the solution of the ILP problem (5)-(6) $\tilde{\boldsymbol{y}}_1, \ldots, \tilde{\boldsymbol{y}}_J \in \mathbb{N}^n$ is such that $\sum_{i=1}^{J} \sum_{t_h \in H} \tilde{\boldsymbol{y}}_j(t_h) = 0$

Example: a buying/selling process



- Buying/selling process that involves the management of a firm and a malicious investor
  - h_1 models the decision of the firm management to put on the market a bond
  - h_2 models the decision of the firm management to buy a bond from the market
  - h_3 models the decision of the firm management to buy and retire a bond from the market
  - l_1 and l_2 model the malicious investor buying and selling a bond, respectively
  - the α and γ markings represent the bonds owned by the firm and by the malicious investor, respectively
  - β markings represent the bonds available on the market
- The system is NOT SNNI → the malicious investor can infer if the firm management is selling or buying bonds, therefore to get information about how the market and prices will evolve in the future

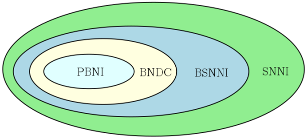
Example: concurrent use of shared resources



Revised version of the example taken from

J. W. Bryans, M. Koutny and P. Y. A. Ryan,
Modelling dynamic opacity using Petri nets with silent actions,
IFIP World Computer Congress, 2004

Various notions of non-interference



- SNNI is not the only non-interference notion that has been given
- SNNI fails to capture the so-called *negative information flows* (an intruder can infer that a high-level event will *never occur anymore*)
- Bisimulation SNNI (BSNNI) is a more restrictive notion that permits to capture

  also negative information flows
  - The BSNNI case for unlabeled bounded PN systems has been tackled in

    F. Basile and G. De Tommasi
    Assessment of bisimulation non-interference in discrete event systems modelled with bounded Petri nets,
    IEEE Control Systems Letters, 2021
- Structural notions of non-interference such as Place-based non-interference (PBNI)
- Intransitive non-interference (INI) enables the specification of a set of security domains and defines which one are allowed to interfere

  N. B. Hadj-Alouane *et .al*
  On the Verification of Intransitive Noninterference in Mulitlevel Security,
  IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics, 2005

  P. Baldan and A. Beggiato
  Multilevel transitive and intransitive non-interference, causally,
  Theoretical Computer Science, 2018
- **Although is the *simplest* (and more permissive) *type* of non-interference, SNNI represents the starting point for any of the *advanced* concepts**

The *opacity* problem

- **Opacity** is another important property of information flow in systems modeled as DES
- When dealing with opacity the secret can be either
  - a system state (initial, current, final) → *State-based opacity*
  - a sequence of events → *Language-based opacity (LBO)*
- In DES context opacity has recently gained a lot of interest
- Opacity is expressive as SNNI but is not comparable to BSNNI and other non-interference concepts
- Equivalence between SNNI and opacity has been proved in

J. Bryans, M. Koutny, L. Mazaré, and P. Ryan
Opacity generalised to transition systems,
International Journal of Information Security, 2008

A. Saboori and C. Hadjicostis
Opacity-enforcing supervisory strategies via state estimator constructions
IEEE Transactions on Automatic Control, 2011

Projection operator in the context of opacity

- In the context of opacity the set of events is partitioned as

$$E = E_o \cup E_{uo} \,, \quad E_o \cap E_{uo} = \emptyset$$

- The usual projection on E_o is defined
  - $Pr(\varepsilon) = \varepsilon$
  - $Pr(\sigma e) = Pr(\sigma)e \quad \text{if } e \in E_o \, \mathrm{Pr}(\sigma) \quad \text{otherwise}$

Initial-State Opacity (ISO)

ISO

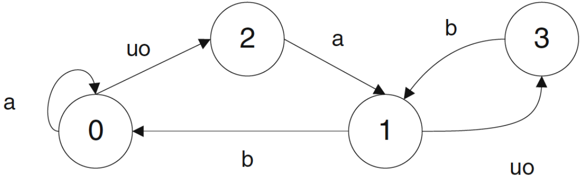
Given a system $G = (X, E, f, X_0)$, a projection Pr, a set of secret initial states $X_S \subseteq X_0$, and set of non-secret initial states $X_{NS} \subseteq X_0$, G is **initial-state opaque** if $\forall\ \bar{x} \in X_S$ and $\forall\ \sigma \in \mathcal{L}(G, \bar{x})$, $\exists\ \bar{y} \in X_{NS}$ and $\exists\ \sigma' \in \mathcal{L}(G, \bar{y})$, such that

$$Pr(\sigma) = Pr(\sigma')$$

Example ISO

$E_{uo} = \{uo\}$, secret state $X_S = \{2\}$

non-secret states $X_{NS} = X \setminus X_S$



Current-State Opacity (CSO)

CSO

Given system $G = (X, E, f, X_0)$, a projection Pr, a set of secret states $X_S \subseteq X$, and set of non-secret states $X_{NS} \subseteq X$, G is **current-state opaque** if $\forall\, \bar{x} \in X_0$ and $\forall\, \sigma \in \mathcal{L}(G, \bar{x})$ such that $f(\bar{x}, \sigma) \in X_S$, $\exists\, \bar{y} \in X_0$, $\exists\, \sigma' \in \mathcal{L}(G, \bar{y})$ such that

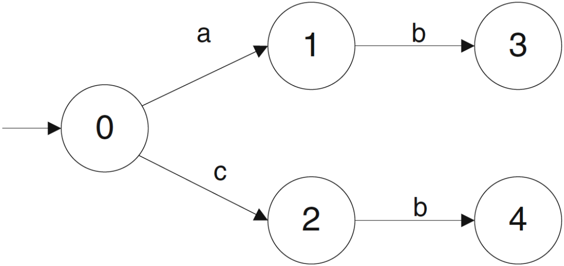
i) $f(\bar{y}, \sigma') \in X_{NS}$

ii) $Pr(\sigma) = Pr(\sigma')$

Example CSO

$$E_{uo} = \{a, b\}\ , \qquad \text{secret state } X_S = \{3\}$$

$$\text{non-secret states } X_{NS} = X \setminus X_S$$



Initial-and-Final State Opacity (IFO)

IFO

Given system $G = (X, E, f, X_0)$, projection Pr, set of secret state pairs $X_{sp} \subseteq X_0 \times X$, and set of non-secret state pairs $X_{nsp} \subseteq X_0 \times X$, G is **initial-and-final-state opaque** if $\forall\ (x_0, x_f) \in X_{sp}$ and $\forall\ \sigma \in \mathcal{L}(G, x_0)$ such that $f(x_0, \sigma) = x_f$, there is a pair $(y_0, y_f) \in X_{nsp}$ and a string $\sigma' \in \mathcal{L}(G, y_0)$ such that

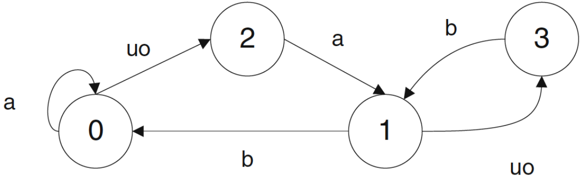
i) $f(y_0, \sigma) = y_f$

ii) $Pr(\sigma) = Pr(\sigma')$

Example IFO

$E_{uo} = \{uo\}$,   secret pair $X_{sp} = \{(3\,,1)\}$

non-secret pairs $X_{nsp} = \{(1\,,0)\,,(1\,,1)\,,(1\,,2)\,,(1\,,3)\}$



Language-based Opacity (LBO)

LBO

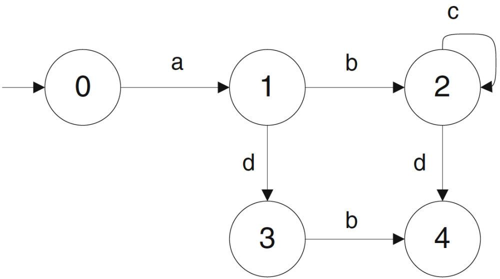
Given system $G = (X, E, f, X_0)$, a projection Pr, a secret language $L_S \subseteq \mathcal{L}(G, X_0)$, and non-secret language $L_{NS} \subseteq \mathcal{L}(G, X_0)$, G is **language-based opaque** if for every string $\sigma \in L_S$, there exists another string $\sigma' \in L_{NS}$ such that

$$Pr(\sigma) = Pr(\sigma')$$

or, equivalently

$$L_S \subseteq Pr^{-1}\left(Pr(L_{NS})\right)$$

Example LBO



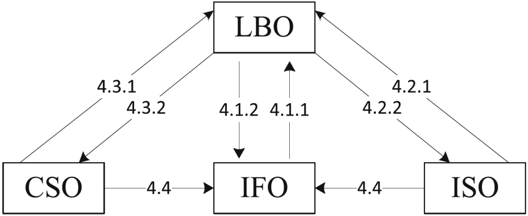
$E_o = \{a\,,b\,,c\}$, secret language $L_S = \{abd\}$

non-secret language $L_{NS} = \{abc^*d\,,adb\}$

secret language $L_S = \{abcd\}$, non-secret language $L_{NS} = \{adb\}$

not opaque

“All opacities are the same"



The equivalence for the case of FSM has been proved in

Y.-C. Wu and S. Lafortune,
Comparative analysis of related notions of opacity in centralized and coordinated architectures,
Discrete Event Dynamic Systems: Theory and Applications, 2013

where also proper transformations have been defined

Verification of CSO for FSM

- Build the **observer automaton** $Obs(G, X_0)$
- The state of $Obs(G, X_0)$ reached by $\sigma \in Pr(\mathcal{L}(G, X_0))$ represents the intruder's state estimate after observing σ
- To verify CSO: examine all reachable states in $Obs(G, X_0) \rightarrow$ **The system is CSO if no state in $Obs(G, X_0)$ contains secret states but not non-secret states**
- When constructing the observer no assumption are made on the set of secret states → no reconstruction of the observer is required if the set of secret states changes

Verification of LBO for regular languages

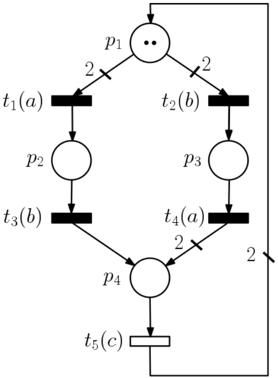
- Build G_1 and G_2 that **mark** L_S and L_{NS}, respectively
- Construct the correspondent observers G_1^{obs} and G_2^{obs}
- **If**

$$\mathcal{L}_m\left(G_1^{obs}\right) = \mathcal{L}_m\left(G_1^{obs} \times G_2^{obs}\right) \tag{7}$$

  then the system is LBO
- Condition (7) implies that

$$Pr\left(L_S\right) \subseteq Pr(L_{NS})$$

The PNs case: a toy example



- the secret sequence is *abc*
- *c* is the only observable event (whose occurrence can be directly *measured*)
- observing the single occurrence of *c*, an intruder will never no if either *abc* or *bac* occurred
- the system is said to be opaque

Assess LBO in PN systems

- Labeled PN systems
- Exploit (once again) the algebraic representation of PN systems
- Assess opacity by solving ILP problems
- **Main assumptions**
  - The secret language L_S has finite cardinality
    - the *non-secret language* is assumed to be equal to $L_{NS} = \mathcal{L} \setminus L_S$
  - The unobservable subnet is *acyclic*
    - prevents the occurrence of arbitrarily long sequences of unobservable events (which in turn would prevent an intruder to detect the occurrence of a secret for an arbitrarily long period)
- *Unnecessary assumptions*
  - the system does not need to be bounded
  - the initial marking is not given ($\boldsymbol{m}_0$ is assumed uncertain, i.e. $\boldsymbol{m}_0$ belongs to a set $\mathcal{M}_0$)

Notation (I)

- The labeling function: $\ell : T \mapsto E$
- Labeled PN system (LPN): $\mathcal{G}\langle N\,, \mathcal{M}_0\,, \ell\rangle$
- Language generated by the LPN: $\mathcal{L}(\mathcal{G}\,, \mathcal{M}_0)$
- Secret language **(assumed to be finite)**: $L_S \subset \mathcal{L}(\mathcal{G}\,, \mathcal{M}_0)$
- Set of transitions associated with the event e:
  $T^e = \{t \in T \mid \ell(t) = e\,, \text{with } e \in E\}$
- Length of a word $w \in E^*$: $|w|$
- Occurrences of $e \in E$ in $w \in E^*$: $|w|_e$
- i-th event in the word w: $w[i]$

Notation (II)

- Observable and unobservable transitions:

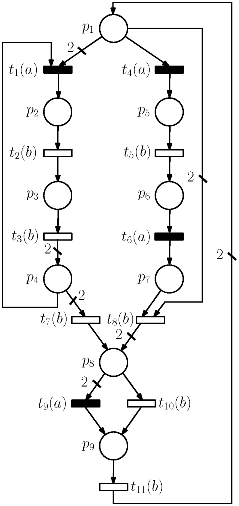
$$T_o = \{t \in T \mid \ell(t) \in E_o\} \ ,$$
$$T_{uo} = \{t \in T \mid \ell(t) \in E_{uo}\} \ ,$$

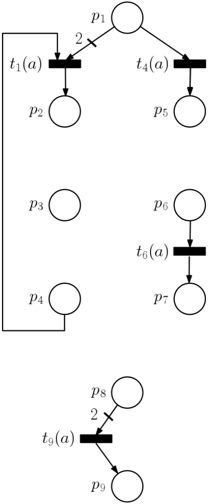
- Given a **firing count vector** $\boldsymbol{\sigma} \in \mathbb{N}^n$, we would like to consider only the firings of either the observable or the unobservable transitions. Hence the following notation is introduced:

$$\boldsymbol{\sigma}_{|T_o} \in \mathbb{N}^n \,, \ \text{with } \boldsymbol{\sigma}_{|T_o}(t) = \begin{cases} \boldsymbol{\sigma}(t) & \text{if } t \in T_o \\ 0 & \text{if } t \notin T_o \end{cases}$$

$$\boldsymbol{\sigma}_{|T_{uo}} \in \mathbb{N}^n \,, \ \text{with } \boldsymbol{\sigma}_{|T_{uo}}(t) = \begin{cases} \boldsymbol{\sigma}(t) & \text{if } t \in T_{uo} \\ 0 & \text{if } t \notin T_{uo} \end{cases}$$

Unobservable subnet





A secret word

$$w \; = \; w_{uo}^{1} e_{o}^{1} w_{uo}^{2} e_{o}^{2} \cdots w_{uo}^{\rho} e_{o}^{\rho} \, ,$$

where:

- $w_o \; = \; Pr(w) = e_o^1 \cdots e_o^\rho$
- unobservable subwords w_{uo}^{i}, with $i = 1 \, , \ldots \, \rho$, may also be empty.

An algebraic characterization of LBO (I)

$$\boldsymbol{\mu} = \boldsymbol{m}_{0_1} \circ (\mu_1 * \mathbf{1}) + \ldots + \boldsymbol{m}_{0_\mathbf{M}} \circ (\mu_\mathbf{M} * \mathbf{1}) \,, \tag{8}$$

$$\sum_{i=1}^{M} \mu_i = 1 \,, \tag{9}$$

$$\boldsymbol{c}^i = \sum_{t^j \in T^{e_o^i}} \boldsymbol{C}(\cdot\,, t^j) \circ (\gamma_{ij} * \mathbf{1}) \,, \ \forall\, \mathbf{i} = \mathbf{1}\,, \ldots\,, \rho \,, \tag{10}$$

$$\sum_{j=1}^{\mathrm{card}\left(T^{e_o^i}\right)} \gamma_{ij} = 1 \,, \qquad \forall\, i = 1\,, \ldots\,, \rho \,, \tag{11}$$

$$\begin{aligned}
&\boldsymbol{\mu} + \boldsymbol{C}_{uo} \cdot \boldsymbol{\sigma}_{1_{|T_{uo}}} \geq \mathbf{0} \,, \\
&\boldsymbol{\mu} + \boldsymbol{C}_{uo} \cdot \boldsymbol{\sigma}_{1_{|T_{uo}}} + \boldsymbol{c}^1 \geq \mathbf{0} \,, \\
&\ldots \\
&\boldsymbol{\mu} + \boldsymbol{C}_{uo} \cdot \sum_{i=1}^{\rho} \boldsymbol{\sigma}_{i_{|T_{uo}}} + \sum_{i=1}^{\rho-1} \boldsymbol{c}^i \geq \mathbf{0} \,, \\
&\boldsymbol{\mu} + \boldsymbol{C}_{uo} \cdot \sum_{i=1}^{\rho} \boldsymbol{\sigma}_{i_{|T_{uo}}} + \sum_{i=1}^{\rho} \boldsymbol{c}^i \geq \mathbf{0} \,,
\end{aligned} \tag{12}$$

- (1) and (2) permit to select one over the *M* possible initial markings
- (3) and (4) associate the firing of single transition for each observable event e_o^i in the secret word *w*
- (5) are the constraints that must be satisfied by the firing count vectors of the *explanations* of $w_o = Pr(w)$

An algebraic characterization of LBO (II)

$$\sum_{t\in T^{e_{uo_k}}}\sum_{i=1}^{\rho}\boldsymbol{\sigma}_{i_{|T_{uo}}}(t) - |w|_{e_{uo_k}} + 1 \leq B\cdot(1-\delta_{k1})\,, \qquad \forall\, e_{uo_k}\in E_{uo}\,, \tag{13}$$

$$\sum_{t\in T^{e_{uo_k}}}\sum_{i=1}^{\rho}\boldsymbol{\sigma}_{i_{|T_{uo}}}(t) - |w|_{e_{uo_k}} \geq -B\cdot\delta_{k1}\,, \qquad \forall\, e_{uo_k}\in E_{uo}\,, \tag{14}$$

$$-\sum_{t\in T^{e_{uo_k}}}\sum_{i=1}^{\rho}\boldsymbol{\sigma}_{i_{|T_{uo}}}(t) + |w|_{e_{uo_k}} + 1 \leq B\cdot(1-\delta_{k2})\,, \qquad \forall\, e_{uo_k}\in E_{uo},\,, \tag{15}$$

$$-\sum_{t\in T^{e_{uo_k}}}\sum_{i=1}^{\rho}\boldsymbol{\sigma}_{i_{|T_{uo}}}(t) + |w|_{e_{uo_k}} \geq -B\cdot\delta_{k2}\,, \qquad \forall\, e_{uo_k}\in E_{uo}\,, \tag{16}$$

$$\delta_{k1}+\delta_{k2}\leq 1\,, \qquad \forall\, k=1\,,\ldots\,,\mathrm{card}(E_{uo})\,, \tag{17}$$

$$\sum_{k=1}^{\mathrm{card}(E_{uo})}(\delta_{k1}+\delta_{k2})\geq 1\,. \tag{18}$$

- In order to have opacity, what we want is that $\sum_{t\in T^{e_{uo_k}}}\sum_{i=1}^{\rho}\boldsymbol{\sigma} i_{|T_{uo}}(t)$ is different from $|w|_{e_{uo_k}}$ for at least one unobservable event e_{uo_k}
- Exploiting the technique proposed in Bemporad and Morari 1999, (6)-(11) have been added to force the firing count vectors of the explanations to have at least one component different from the firing count vector of the unobservable substring in the secret

A. Bemporad and M. Morari,
Control of systems integrating logic, dynamics, and constraint,
Automatica, vol. 35, no. 3, pp. 407–427, 1999

Sufficient condition

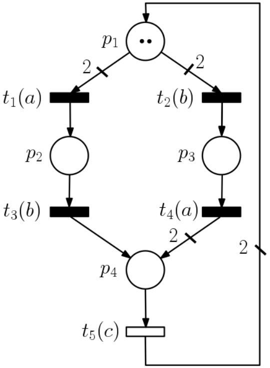
Lemma

Let $\mathcal{G} = \langle N, \mathcal{M}_0, \ell \rangle$ be a labeled net system, $w \in \mathcal{L}_S$ a secret word such that $|w_o| = \rho$, with $w_o = Pr(w) = w_o = e_o^1 \cdots e_o^\rho$, and B be a sufficiently large integer. If the set of constraints (8)–(18) admits a solution, then there exists at least one $w' \in \mathcal{L}(\mathcal{G}, \mathcal{M}_0)$ such that $Pr(w') = Pr(w)$.

Theorem

Let $\mathcal{G} = \langle N, \mathcal{M}_0, \ell \rangle$ be a labeled net system and $\mathcal{L}_s \subseteq \mathcal{L}(\mathcal{G}, \mathcal{M}_0)$ a finite secret language. If for all $w \in \mathcal{L}_s$ the set of constraints (8)–(18) admits a solution, then $\mathcal{G}$ is LBO.

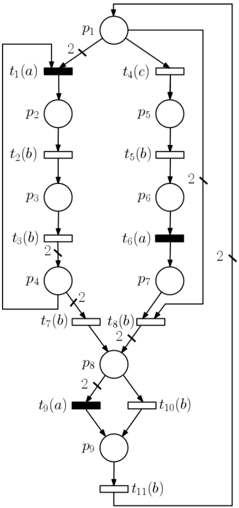
Conservativeness of the sufficient condition



- The proposed sufficient condition cannot take into account the order of the unobservable events in each unobservable subword of the secret
- At the expense of an increase of the number of optimization variable (hence of the computational burden), a necessary and sufficient condition can be derived

F. Basile and G. De Tommasi
An algebraic characterization of language-based opacity in labeled Petri,
14th *International Workshop on Discrete Event Systems*, 2018

Example



- $L_S = \{abb\}$
-

$$\begin{aligned}\mathcal{M}_0'' &= \{\boldsymbol{m}_{0_1}'' , \boldsymbol{m}_{0_2}''\} \\ &= \Big\{ \begin{pmatrix}2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0\end{pmatrix}^T , \\ &\quad \begin{pmatrix}2 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0\end{pmatrix}^T \Big\} .\end{aligned}$$

- The sufficient condition requires to check the feasibility problem only for one word
  - The feasibility problem admits a solution, since *bb* is enabled under $\boldsymbol{m}_0''$

References

- F. Basile, G. De Tommasi and C. Sterle,
  Non-interference enforcement via supervisory control in bounded Petri nets,
  IEEE Transactions on Automatic Control, to appear 2021
- Y.-C. Wu and S. Lafortune,
  Comparative analysis of related notions of opacity in centralized and coordinated architectures,
  Discrete Event Dynamic Systems: Theory and Applications, 2013

Security issues in DES: non-interference and opacity

From observability to privacy and security in discrete event systems

Prof. Gianmaria De Tommasi
Email: detommas@unina.it

December 2020

DIETI. UNIVERSITA' DEGLI STUDI DI NAPOLI FEDERICO II
DIPARTIMENTO DI INGEGNERIA ELETTRICA E DELLE TECNOLOGIE DELL'INFORMAZIONE