On-line verification of current-state opacity by Petri nets and integer linear programming

Publisher version https://doi.org/10.1016/j.automatica.2018.04.021

On-line Verification of Current-State Opacity by Petri Nets and Integer Linear Programming ⋆

Xuya Cong [a,b], Maria Pia Fanti [c], Agostino Marcello Mangini [c], Zhiwu Li [d,a]

[a] *School of Electro-Mechanical Engineering, Xidian University No. 2 South Taibai Road, Xi'an 710071, China*

[b] *Key Laboratory of Electronic Equipment Structure Design (Xidian University), Ministry of Education, Xi'an 710071, China*

[c] *Department of Electrical and Information Engineering, Polytechnic of Bari, 70125 Bari, Italy*

[d] *Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau*

Abstract

Opacity is a security and privacy property that evaluates whether an external observer (intruder) can infer a *secret* of a system by observing its behaviour. This paper proposes an on-line approach to address the problem of current-state opacity in discrete event systems modeled in a labeled Petri Net (PN) framework and by observing its evolution. An observation of the system is said to be *current-state opaque* if an intruder is unable to determine whether the current-state of the system is within a set of secret states, otherwise it is said to be not current-state opaque. The proposed approach to verify the current-state opacity works on-line: the intruder waits for the occurrence of an observable event and uses Integer Linear Programming problem solutions to verify if the given observation of the system is current-state opaque. Moreover, the proposed method is applied in two different settings: i) a centralized approach where the intruder has full knowledge of the system model but is able to partially observe the system behaviour; ii) a decentralized approach where a set of intruders can observe different event sets and collaborate with a coordinator to check the same secret. Finally, several examples are presented to demonstrate the efficiency of the proposed method.

Key words: On-line security analysis; Petri nets; Integer linear programming; Decentralized systems.

1 Introduction

The problems of security and privacy have received extensive concerns in on-line services of networked and cyber-physical systems over the last few decades. To formulate these problems, various notions of security and privacy have been proposed in the literature, such as anonymity [23], [29], non-interference [7], [17] and opacity [5], [24], [25]. In particular, opacity is a security and privacy property that evaluates whether an external observer (intruder) can infer a *secret* of a system by observing its behaviour. Depending on the definition of the secret, there are two main kinds of opacity properties provided in the related literature: language-based opacity and state-based opacity. State-based opacity defines the secret as a set of secret states and it can be further classified as initial-state opacity [25], [27], [30], initial-and-final-state opacity [33], current-state opacity [28], [31] and k-step opacity [26]. The work in [33] proposes a polynomial algorithm to transform one of the four kinds of opacity (language-based, initial-state, initial-and-final-state, and current-state) to any other.

This paper focuses on the current-state opacity. In particular, a system is said to be current-state opaque with respect to a secret, if for any observation, the intruder cannot infer that the current state of the system belongs to the secret. More precisely, opacity requires that for any observation the intruder estimates the set of states consistent with the observation and verifies that is not subset of the secret. Some works address the verification of current-state opacity in Discrete Event Systems (DESs) that are modeled as Finite-State Automata (FSA). In this framework, the intruder is considered as an external observer that has full knowledge of the struc-

⋆ This paper was not presented at any IFAC meeting. Corresponding author Zhiwu Li.

Email addresses: congxuya@163.com (Xuya Cong), mariapia.fanti@paliba.it (Maria Pia Fanti), agostinomarcello.mangini@poliba.it (Agostino Marcello Mangini), zhwli@xidian.edu.cn (Zhiwu Li).

ture of the system but has only partial observation on its events. The well-known approach to verify current-state opacity is to build an observer automaton [24] that represents the intruder's state estimate after a word is observed. Motivated by the work in [24], Saboori and Hadjicostis [28] also present current-state opacity notions in Probabilistic Finite Automata (PFA) and the corresponding verification methods. However, the construction of the observer has $O(2^n)$ state-space and time complexity where n is the number of the states in the automaton.

Most opacity-related studies consider a centralized architecture in the framework of automata, i.e., only one intruder verifies the opacity of the system. Due to distributed nature of real systems, several works also take into account distributed definitions of opacity. The study in [1] considers several intruders that have different observation masks and secrets. A system is said to be concurrently opaque if all the secrets are safe. Wu and Lafortune [33] extend the opacity notion to a coordinated architecture where multiple intruders work together with a coordinator to discover the same secret. In their work, joint opacity properties are first introduced and adapted to the coordinated architecture. In addition, Paoli and Lin [21] introduce decentralized opacity definitions for the cases with and without coordination among agents based on languages.

Compared with automata, PNs have the advantages of modeling DESs by their twofold representation: graphical and mathematical. Thus, PNs have been extensively applied to structural analysis theory [18], deadlock control [19], [32], supervisory control theory [20] and scheduling [2].

In the framework of PNs, current-state opacity is first introduced in [5] and this notion is then extended to labeled transition systems in [6]. Moreover, Tong *et al.* [31] solve the verification of current-state opacity in labeled Petri Nets (LPNs). The work in [31] proposes a necessary and sufficient condition for current-state opacity by using the notion of basis markings [8], [16]. The advantage of this method is to avoid the exhaustive enumeration of the reachable markings. However, it can be only applied to bounded PNs and a large memory may be still required.

In order to avoid the states enumeration of a system, this paper presents an on-line verification method of current-state opacity by employing LPN models and Integer Linear Programming (ILP) problem, an approach also used to solve the on-line fault diagnosis [3], [13], [14] and fault diagnosability tests [4]. More precisely, the structure of the LPN and the initial marking are known by the intruder which only has partial observation of the transitions. The intruder waits for an observable event and exploits an algorithm to determine whether the system behaviour remains in the secret or not. By the definition of current-state opacity, if there exists an observation such that the intruder can decide that all the markings consistent with the observation belong to the secret, then the system is said to be not current-state opaque with respect to the secret. Moreover, the proposed technique is also extended to the decentralized architecture that is composed by a number of local intruders communicating their own output information with a coordinator. Each local intruder has a full knowledge of the net structure and its initial marking, but it observes only a subset of the observable events. The coordinator is used to produce the global result as the single *global (system) intruder*. To this aim, we propose a protocol for the communication between the local intruders and the coordinator.

As a conclusion of this section, we summarize the main features and contributions of this paper.

(1) An on-line algorithm for opacity is presented in the framework of PN system exploiting ILP. The algorithm checks the current-state opacity property for each given observation of the system by avoiding off-line computation of the observer.
(2) By using the on-line strategy, the proposed method avoids the redesign and redefinition of the intruder when the system structure changes.
(3) The on-line method for verification of current-state opacity is applied in a decentralized architecture.
(4) The proposed methods in centralized and decentralized architecture are general since they both can be applied to the net with bounded and unbounded state space.

The rest of this paper is organized as follows. Section 2 briefly introduces some basics of the PN formalism. Section 3 defines the intruder, proposes the on-line algorithm to verify current-state opacity and uses some examples to show the efficiency of the approach. Section 4 extends the algorithm proposed in Section 3 to a decentralized architecture and an example illustrates this distributed approach. Finally, Section 5 draws the conclusion.

2 Preliminaries

2.1 Petri nets

This section reviews some basics of PNs [22] used in the paper.

A PN is a 4-tuple $PN = (P, T, Pre, Post)$, where P is a set of m places represented by circles, T is a set of n transitions represented by bars, $Pre : P \times T \to \mathbb{N}$ and $Post : P \times T \to \mathbb{N}$ are the pre- and post-incidence matrices, respectively, which specify the arcs connecting places and transitions. More precisely, for each $p \in P$ and $t \in T$ element $Pre(p,t)$ $(Post(p,t))$ is equal to a

natural number indicating the arc multiplicity if an arc going from p to t (from t to p) exists, and it is equal to 0 otherwise. Note that $\mathbb{N}$ is the set of non-negative integers. Matrix $C = Post - Pre$ is the $m \times n$ incidence matrix of the PN.

The state of a PN is given by its current marking that is a mapping $M : P \to \mathbb{N}^m$, assigning to each place an integer number of tokens. The marking of place p is denoted by $M(p)$. For simplicity, markings can also be denoted as $M = \sum_{p \in P} M(p) \cdot p$. A PN system $\langle PN, M_0 \rangle$ is a net PN with an initial marking M_0.

A transition $t_j \in T$ is enabled at M if $M \geq Pre(\cdot, t_j)$ holds and $M[t_j\rangle$ is used to denote that $t_j \in T$ is enabled at marking M. When fired, t_j produces a new marking M', denoted by $M[t_j\rangle M'$ that is computed by the PN state equation $M' = M + C \cdot \overrightarrow{t_j}$, where $\overrightarrow{t_j}$ is an n-dimensional firing vector corresponding to the jth canonical basis vector.

Let $\sigma = t_1 t_2 \dots t_k$ be a sequence of transitions (firing sequence) and let k be its length, given by the number of transitions that σ contains. The fact that a transition $t \in T$ appears in the sequence σ is denoted by $t \in \sigma$. Moreover, the notation $M[\sigma\rangle$ denotes that σ is enabled at M and $M[\sigma\rangle M'$ denotes that the firing of σ yields M'. The set of all sequences that can fire in a net system $\langle PN, M_0 \rangle$ is denoted by $L(PN, M_0) = \{\sigma \in T^* | M_0[\sigma\rangle\}$. In addition, $\vec{\sigma} : T \to \mathbb{N}^n$ is the firing vector associated with a sequence σ.

A marking M is said to be reachable from $\langle PN, M_0 \rangle$ if there exists a firing sequence σ such that $M_0[\sigma\rangle M$. The set of all markings reachable from M_0 defines the reachability set of $\langle PN, M_0 \rangle$, which is denoted as $R(PN, M_0)$.

A PN having no directed cycles is said to be *acyclic*. The following theorem shows an important property of this subclass of PNs.

Theorem 1 *[11] Let $\langle PN, M_0 \rangle$ be an acyclic PN.*

(1) If vector y satisfies equation $M_0 + C \cdot y \geq \vec{0}$, there exists a firing sequence σ fireable from M_0 such that $\vec{\sigma} = y$.
(2) A marking M is reachable from M_0 iff there exists a non-negative integer solution y satisfying the state equation $M = M_0 + C \cdot y$.

2.2 Labeled Petri nets

An LPN is 4-tuple $G = (PN, M_0, E, \lambda)$ where $\langle PN, M_0 \rangle$ is a PN system, E is an alphabet (a set of labels) and $\lambda : T \to E \cup \{\varepsilon\}$ is a labeling function that assigns to each transition $t \in T$ either a symbol $e \in E$ or the empty word ε.

We assume that the intruder has complete knowledge of the net system but partial observation of its behaviour. Namely, the set of transitions can be partitioned into $T = T_o \cup T_u$ with $T_o \cap T_u = \emptyset$, where T_o (resp. T_u) is the set of $|T_o| = n_o$ (resp. $|T_u| = n_u$) observable (resp. unobservable) transitions whose occurrence can (resp. cannot) be detected by the intruder. Hence, the labeling function λ is defined as follows: if $t \in T_o$ then $\lambda(t) = e \in E$, and if $t \in T_u$ then $\lambda(t) = \varepsilon$. Here, we assume that the same label $e \in E$ can be associated to more than one transition. In the following, we denote by $T(e) = \{t \in T_o | \lambda(t) = e\}$ the set of transitions associated with the same label $e \in E$.

Moreover, we denote as w the sequence of events associated with the sequence $\sigma \in T^*$ such that $w = \lambda(\sigma)$ by using the extended form of the labeling function $\lambda : T^* \to E^*$. The set of languages generated by an LPN is denoted as $\mathcal{L}(PN, M_0) = \{w \in E^* | \exists \sigma \in L(PN, M_0) : \lambda(\sigma) = w\}$. In addition, we denote by $\sigma_u \in \sigma$ ($\sigma_o \in \sigma$) the subsequence of σ composed of the unobservable (observable) transitions and by $\vec{\sigma}_u : T_u \to \mathbb{N}^{n_u}$ ($\vec{\sigma}_o : T_o \to \mathbb{N}^{n_o}$) the corresponding firing vector.

Given a net $PN = (P, T, Pre, Post)$ and a subnet $T_A \subseteq T$ of its transitions, we define the T_A-induced subnet of PN as a new net $PN_A = (P, T_A, Pre_A, Post_A)$ where Pre_A and $Post_A$ are the restrictions of Pre and $Post$ to T_A, i.e., PN_A is the net obtained from PN by removing all transitions in $T \backslash T_A$, which is denoted by $PN_A \angle_{T_A} PN$. In the following, matrices $C_u = Post_u - Pre_u$ and $C_o = Post_o - Pre_o$ denote the restriction of the incidence matrix C to T_u and T_o, respectively.

Let w be an observed word. We define $\mathcal{S}(w) = \{\sigma \in L(PN, M_0) | \lambda(\sigma) = w\}$ as the set of firing sequences consistent with w and $\mathcal{C}(w) = \{M \in \mathbb{N}^m | \exists \sigma \in \mathcal{S}(w) : M_0[\sigma\rangle M\}$ as the set of markings consistent with w.

3 Verification of current-state opacity in the centralized approach

In this section, we provide an on-line approach for verification of current-state opacity in the centralized approach.

3.1 Description of current-state opacity

In the following, we first recall opacity definitions in [31] for DESs that are modeled by LPNs. A secret is defined as a set of secret states $S \subseteq R(PN, M_0)$.

Definition 1 *[31] Let G be an LPN system and S be a secret. An observation w of G is said to be current-state opaque wrt S if $\mathcal{C}(w) \nsubseteq S$ holds.*

Based on Definition 1, the current-state opacity definition for a system is given as follows.

Definition 2 *[31] Let G be an LPN system and S be a secret. G is said to be current-state opaque wrt S if all observations w are current-state opaque wrt S.*

Motivated by Definitions 1 and 2, we provide the following two definitions of a not current-state opaque observation and a not current-state opaque system, respectively.

Definition 3 *Let G be an LPN system and S be a secret. An observation w of G is said to be not current-state opaque wrt S if $\mathcal{C}(w) \subseteq S$ holds.*

A not current-state opaque observation w implies that the intruder can infer that all the markings consistent with w belong to the secret, i.e., $\forall M \in \mathcal{C}(w) : M \in S$.

Consequently, a not current-state opaque system is defined as follows.

Definition 4 *Let G be an LPN system and S be a secret. G is said to be not current-state opaque wrt S if there exists an observation w that is not current-state opaque wrt S.*

In this paper, the following set of Generalized Mutual Exclusion Constraints (GMECs) [15] describes the secret:

$$S = \bigcap_{q=1}^{r} \{M \in \mathbb{N}^m | x_q^T \cdot M \leq k_q\},$$

where $x_q \in \mathbb{Z}^m$ and $k_q \in \mathbb{Z}$ with $q = 1, 2, \ldots, r$. Note that $\mathbb{Z}$ is the set of integers. Such a set of GMECs (x_q, k_q) is denoted as $S = \{M \in \mathbb{N}^m | X \cdot M \leq K\}$, where $X = [x_1, x_2, \ldots, x_r]^T$ and $K = [k_1, k_2, \ldots, k_r]^T$.

3.2 The on-line intruder specification

In this subsection, given an observed word w, we show how to characterize the sets $S(w)$ and $C(w)$ by solving ILP problems and we specify the on-line intruder.

Firstly, the following assumption is given for the system under investigation:

A1) The T_u-induced subnet $PN_u \angle_{T_u} PN$ and T_o-induced subnet $PN_o \angle_{T_o} PN$ are acyclic.

In particular, assumption A1 allows us to study the reachability of the unobservable and observable subnets by using the state equation. The inputs of the intruder are the LPN system $G = (PN, M_0, E, \lambda)$, the secret S modeled by a set of GMECs, and the observed word $w \in \mathcal{L}(PN, M_0)$. The output of the intruder is the set-valued function $\Phi(w)$ that is defined as follows:

Definition 5 *An on-line intruder is a function $\Phi : \mathcal{L}(PN, M_0) \to \{Y, N\}$ that associates to each observation $w \in \mathcal{L}(PN, M_0)$ the following sets:*

(1) $\Phi(w) = \{Y\}$ if the observation of the system is current-state opaque wrt the secret S.
(2) $\Phi(w) = \{N\}$ if the observation of the system is not current-state opaque wrt the secret S.

Given an LPN system $G = (PN, M_0, E, \lambda)$ with language $\mathcal{L}(PN, M_0)$ and satisfying assumption A1, we specify an intruder that works on-line and determines whether an observation is opaque or not after the occurrence of each new event. More precisely, for each initial marking $M_0 \in \mathbb{N}^m$, at the occurrence of an observed word $w \in \mathcal{L}(PN, M_0)$, the following proposition shows a linear algebraic characterization of each transition sequence $\sigma \in T^*$ whose firing at M_0 is consistent with the observation $w = \lambda(\sigma)$.

Proposition 1 *Let $\mathcal{L}(PN, M_0)$ be the language of an LPN system $G = (PN, M_0, E, \lambda)$ satisfying assumption A1. Given a word $w \in \mathcal{L}(PN, M_0)$ denoted by $w = e_1 e_2 \ldots e_h$ (where $e_i \in E$ for $i = 1, 2, \ldots, h$ is the ith observed event), there exists at least one sequence $\sigma = \sigma_{u_1} \sigma_{o_1} \sigma_{u_2} \sigma_{o_2} \ldots \sigma_{u_h} \sigma_{o_h} \sigma_{u_{h+1}}$ with $|\sigma_{u_i}| \geq 0$ for $i = 1, 2, \ldots, h+1$ and $|\sigma_{o_i}| = 1$ for $i = 1, 2, \ldots, h$ enabled at the initial marking M_0 such that $\lambda(\sigma) = w = e_1 e_2 \ldots e_h$* iff *there exist $2h+1$ firing vectors $\vec{\sigma}_{u_1}, \vec{\sigma}_{u_2}, \ldots, \vec{\sigma}_{u_{h+1}}, \vec{\sigma}_{o_1}, \vec{\sigma}_{o_2}, \ldots \vec{\sigma}_{o_h}$ that satisfy the following set of constraints denoted by $\rho(M_0, w)$:*

$$\begin{cases} \vec{\sigma}_{u_i} \in \mathbb{N}^{n_u}, \quad for\ i = 1, \ldots, h+1 & (a) \\ \vec{\sigma}_{o_i} \in \mathbb{N}^{n_o}, \quad for\ i = 1, \ldots, h & (b) \\ C_u \sum_{i=1}^{k} \vec{\sigma}_{u_i} \geq Pre_o \cdot \vec{\sigma}_{o_k} - M_0 - C_o \sum_{i=1}^{k-1} \vec{\sigma}_{o_i}, & \\ for\ k = 1, \ldots, h & (c) \\ M_0 + C_u \sum_{i=1}^{h+1} \vec{\sigma}_{u_i} + C_o \sum_{i=1}^{h} \vec{\sigma}_{o_i} \geq \vec{0} & (d) \\ \sum_{t_j \in T(e_1)} \vec{\sigma}_{o_1}(t_j) = 1 & \\ \sum_{t_j \in T(e_2)} \vec{\sigma}_{o_2}(t_j) = 1 & \\ \ldots & \\ \sum_{t_j \in T(e_h)} \vec{\sigma}_{o_h}(t_j) = 1 & (e) \\ \sum_{t_j \notin T(e_1)} \vec{\sigma}_{o_1}(t_j) = 0 & \\ \sum_{t_j \notin T(e_2)} \vec{\sigma}_{o_2}(t_j) = 0 & \\ \ldots & \\ \sum_{t_j \notin T(e_h)} \vec{\sigma}_{o_h}(t_j) = 0 & \end{cases} \quad (1)$$

Proof: *(Only if)* Assume that $\sigma \in \mathcal{S}(w)$ such that $\sigma = \sigma_{u_1} \sigma_{o_1} \ldots \sigma_{u_h} \sigma_{o_h} \sigma_{u_{h+1}}$ and $M_0[\sigma_{u_1} \sigma_{o_1}\rangle M_1 \ldots M_{h-1}[\sigma_{u_h} \sigma_{o_h}\rangle M_h[\sigma_{u_{h+1}}\rangle M_{h+1}$, where M_i is the marking reached after observable sequence σ_{o_i} ($|\sigma_{o_i}| = 1$) fires for

$i = 1, \ldots, h$ and M_{h+1} is the marking reached after unobservable sequence $\sigma_{u_{h+1}}$ fires. The corresponding firing vectors $\vec{\sigma}_{u_1}, \ldots, \vec{\sigma}_{u_{h+1}}, \vec{\sigma}_{o_1}, \ldots, \vec{\sigma}_{o_h}$ trivially verify the $2h+1$ constraints of (1)(a) and (1)(b).

By the enabling condition, we have:

$$M_{i-1} + C_u \cdot \vec{\sigma}_{u_i} \geq Pre_o \cdot \vec{\sigma}_{o_i} \quad for\ i = 1, \ldots, h. \quad (2)$$

Moreover, by the state equation, the firing vectors $\vec{\sigma}_{u_1}$, $\ldots$, $\vec{\sigma}_{u_h}$, $\vec{\sigma}_{o_1}$, $\ldots$, $\vec{\sigma}_{o_h}$ satisfy the constraints:

$$M_{i-1} + C_u \cdot \vec{\sigma}_{u_i} + C_o \cdot \vec{\sigma}_{o_i} = M_i \quad for\ i = 1, \ldots, h. \quad (3)$$

By writing (2) and (3) for each $i = 1, \ldots, h$ and recursively eliminating all the intermediate markings M_i for $i = 1, \ldots, h$ from the obtained equations, it holds that $C_u \sum_{i=1}^{k} \vec{\sigma}_{u_i} \geq Pre_o \cdot \vec{\sigma}_{o_k} - M_0 - C_o \sum_{i=1}^{k-1} \vec{t}_{o_i}$ for $k = 1, \ldots, h$. In addition, for the marking M_{h+1}, by Theorem 1, we have

$$M_0 + C_u \sum_{i=1}^{h+1} \vec{\sigma}_{u_i} + C_o \sum_{i=1}^{h} \vec{\sigma}_{o_i} \geq \vec{0}. \quad (4)$$

Since $\lambda(\sigma) = w$ with $w = e_1 e_2 \ldots e_h$, at each step, only one transition corresponding to the ith observed event e_i for $i = 1, 2, \ldots, h$ can fire. Hence, constraints (1)(e) hold.

(If) If there exist some firing vectors $\vec{\sigma}_{u_1}, \ldots, \vec{\sigma}_{u_{h+1}}, \vec{\sigma}_{o_1}$, $\ldots$, $\vec{\sigma}_{o_h}$ that satisfy the set of constraints $\rho(M_0, w)$, then there exist a sequence M_1, $\ldots$, M_{h-1}, M_h that satisfies (2) and (3), and M_{h+1} that satisfies (4). By Theorem 1, there exists a sequence $\sigma = \sigma_{u_1}\sigma_{o_1} \ldots \sigma_{u_h}\sigma_{o_h}\sigma_{o_{h+1}}$ that is enabled at M_0, which may fire yielding the evolution $M_0[\sigma_{u_1}\sigma_{o_1}\rangle M_1 \ldots M_{h-1}[\sigma_{u_h}\sigma_{o_h}\rangle M_h[\sigma_{u_{h+1}}\rangle M_{h+1}$. Moreover, $\sum_{t_j \in T(e_i)} \vec{\sigma}_{o_i}(t_j) = 1$ and $\sum_{t_j \notin T(e_i)} \vec{\sigma}_{o_i}(t_j) = 0$ in (1)(e) for $i = 1, \ldots, h$ are congruence condition between transitions having the same label for each observed event in w. Hence, $\lambda(\sigma) = w = e_1 \ldots e_h$. □

Remark 1 *Note that the empty word $w = \varepsilon$ belongs to the set $\mathcal{L}(PN, M_0)$. In this case, according to Theorem 1, there exists at least one sequence $\sigma = \sigma_{u_1}$ with $|\sigma_{u_1}| \geq 0$ enabled at the initial marking M_0 such that $\lambda(\sigma) = w = \varepsilon$* iff *there exists a vector $\vec{\sigma}_{u_1}$ that satisfies the following set of constraints denoted by $\rho(M_0, w)$:*

$$\begin{cases} \vec{\sigma}_{u_1} \in \mathbb{N}^{n_u}, \\ M_0 + C_u \cdot \vec{\sigma}_{u_1} \geq \vec{0}. \end{cases} \quad (5)$$

In general, the solution of the set of constraints $\rho(M_0, w)$ is not a singleton and fully characterizes the two sets $\mathcal{S}(w)$ and $\mathcal{C}(w)$. Actually, constraints (1) imply that $M = M_0 + C_u \sum_{i=1}^{h+1} \vec{\sigma}_{u_i} + C_o \sum_{i=1}^{h}$ belongs to $\mathcal{C}(w)$. In order to verify if the behaviour of the system remains in the secret under the given observation $w \in \mathcal{L}(PN, M_0)$, we have to find a possible solution of (1), i.e., a set of firing vectors leading to a marking that does not belong to the secret. The following theorem proves that such a solution can be obtained by solving the ILP Problem 1 (ILPP 1).

Proposition 2 *Let $G = (PN, M_0, E, \lambda)$ be an LPN system and S be a secret. Given an observed word $w = e_1 e_2 \ldots e_h \in \mathcal{L}(PN, M_0)$, let us define the following ILP problem, ILPP 1:*

$$\begin{cases} z_q = max & x_q^T \cdot M \\ s.t. & \rho(M_0, w) \\ & M = M_0 + C_u \sum_{i=1}^{h+1} \vec{\sigma}_{u_i} + C_o \sum_{i=1}^{h} \vec{\sigma}_{o_i}. \end{cases} \quad (6)$$

An observation w of G is current-state opaque wrt S iff *for a GMEC (x_q, k_q) of the secret, ILPP 1 admits a solution $\vec{\sigma}_{u_1}$, $\vec{\sigma}_{u_2}$, $\ldots$, $\vec{\sigma}_{u_{h+1}}$, $\vec{\sigma}_{o_1}$, $\vec{\sigma}_{o_2}$, $\ldots$, $\vec{\sigma}_{o_h}$ and it holds $z_q > k_q$.*

Proof: Since $w \in \mathcal{L}(PN, M_0)$, according to Proposition 1 the set of constraints $\rho(M_0, w)$ fully describes the whole set $\mathcal{C}(w)$ (the estimation of the intruder).

(Only if) By contradiction, let us assume that w is current-state opaque wrt the secret S and z_q =max $x_q^T \cdot M \leq k_q$ with $q = 1, \ldots, r$ holds for each GMEC in the secret S. Then each marking $M \in \mathcal{C}(w)$ satisfies $X \cdot M \leq K$. That is to say, all the markings in the set of $\mathcal{C}(w)$ belong to the secret. According to Definition 3, w is not current-state opaque wrt the secret S, which contradicts the hypothesis.

(If) If z_q=max $x_q^T \cdot M > k_q$ for a GMEC (x_q, k_q) of the secret S, then there exists a marking $M \in \mathcal{C}(w)$ such that $x_q^T \cdot M > k_q$. This implies that marking M is not in the secret and according to Definition 1, the observation w of G is current-state opaque wrt S and the conclusion holds. □

Remark 2 *Note that the empty word $w = \varepsilon$ belongs to the set $\mathcal{L}(PN, M_0)$. In this case, ILPP 1 can be rewritten as follows:*

$$\begin{cases} z_q = max & x_q^T \cdot M \\ s.t. & \rho(M_0, w) \\ & M = M_0 + C_u \cdot \vec{\sigma}_{u_1}, \end{cases} \quad (7)$$

and $w = \varepsilon$ of G is current-state opaque wrt S iff *for a GMEC (x_q, k_q) of the secret, ILPP 1 admits a solution $\vec{\sigma}_{u_1}$ and it holds $z_q > k_q$.* □

Proposition 2 provides the following sufficient and necessary condition to verify whether an LPN system is not current-state opaque.

Corollary 1 *Let $G = (PN, M_0, E, \lambda)$ be an LPN system and S be a secret. The system is not current-state opaque* iff *there exists an observation w such that ILPP 1 admits a solution with $z_q = max\ x_q^T \cdot M \leq k_q$ for each GMEC (x_q, k_q) of the secret.*

Proof: It follows immediately from Definition 4 and Proposition 2.

3.3 The on-line algorithm to verify current-state opacity

Based on the aforementioned results, we propose Algorithm 1 that the intruder can apply on-line to verify the current-state opacity of a given LPN system. In the following, we discuss the details of Algorithm 1.

Algorithm 1. On-line algorithm specifying the intruder function
Input: $G = (PN, M_0, E, \lambda)$, S
Output: $\Phi(w)$
Step 1. *Initializing the variables of the algorithm*
$w := \varepsilon$, $h := 0$, $\Phi(w) := \emptyset$
Step 2. *Verifying if the observed word w is current-state opaque wrt S*
for $q = 1$ to r **do**
Solve ILPP 1

$$\begin{cases} z_q = max & x_q^T \cdot M \\ s.t. & \rho(M_0, w) \\ & M = M_0 + C_u \sum_{i=1}^{h+1} \vec{\sigma}_{u_i} + C_o \sum_{i=1}^{h} \vec{\sigma}_{o_i} \end{cases}$$

if $z_q > k_q$ **then**
$\Phi(w) := \{Y\}$ **go to Step 3**
end if
end for
$\Phi(w) := \{N\}$ **go to Step 4**
Step 3. *Recording the events*
Wait until an event $e \in E$ occurs
$w := we$, $h := h + 1$, $\Phi(w) := \emptyset$
go to Step 2
Step 4. End

Step 1 initializes the variables of the algorithm where h denotes the length of w.

Step 2 verifies whether w is current-state opaque wrt S: if there exists a GMEC (x_q, k_q) of S such that the objective function value of ILPP 1 $z_q > k_q$, then by Proposition 2, w is current-state opaque wrt S. In this case, the algorithm goes to Step 3 to wait for a new event. Moreover, if ILPP 1 admits an optimal solution with $z_q \leq k_q$ for each of the GMEC (x_q, k_q) of the secret, then by Proposition 2, w is not current-state opaque wrt S. Consequently, according to Corollary 1, the system G is not current-state opaque wrt S.

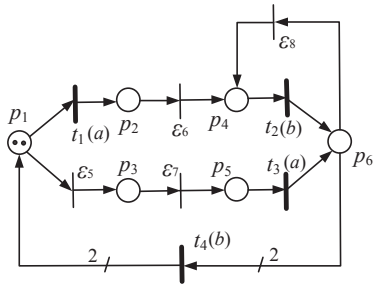

Fig. 1. The LPN system considered in Example 1.

Example 1 *In order to show the application of Algorithm 1, we consider the LPN of a communication system proposed in [31] that is shown in Fig. 1. There are six places, eight transitions and two observable events, i.e., $E = \{a, b\}$. The set of observable transitions is $T_o = \{t_1, t_2, t_3, t_4\}$ such that $T(a) = \{t_1, t_3\}$ and $T(b) = \{t_2, t_4\}$, and the set of unobservable transitions is $T_u = \{\varepsilon_5, \varepsilon_6, \varepsilon_7, \varepsilon_8\}$. The initial marking is $M_0 = [2, 0, 0, 0, 0, 0]^T$. Let the secret be $S = \{M \in \mathbb{N}^6 | X \cdot M \leq K\}$ with $X = [0, 0, 1, 0, 1, 0]$ and $K = 0$. By using the Basis Reachability Graph (BRG) and its observer presented in [31], the system is inferred not current-state opaque wrt the secret. Now, we show the procedure of the proposed on-line algorithm as follows.*

Suppose that no observable event occurs in the system, by Algorithm 1, we obtain $\Phi(\varepsilon) = \{Y\}$, i.e., the observation $w = \varepsilon$ is current-state opaque wrt S.

Assume that the observable event a occurs. By applying Algorithm 1, we infer $\Phi(a) = \{Y\}$, i.e., the observation $w = a$ is current-state opaque wrt S.

Now, assume that the second observable event a occurs: Algorithm 1 provides $\Phi(aa) = \{N\}$, i.e., the observation $w = aa$ is not current-state opaque wrt S. Moreover, according to Definition 4, the system is not current-state opaque wrt S. □

3.4 Computational complexity

As regard the computational complexity of Algorithm 1, we note that the algorithm needs to solve for each observation w at most r ILPPs, which are NP-hard in

theory. To evaluate the computational effort required by the proposed algorithm, we recall that the primary determinants of the computational cost of an ILPP are the numbers of variables and constraints in it. It is easy to infer that the numbers of variables and constraints in each ILPP are $h \cdot n + n_u + m$ and $h \cdot m + 2 \cdot (h + m)$ in the worst case, respectively, where $h \geq 0$ denotes the length of the observation w, n denotes the number of transitions in the LPN, n_u denotes that of unobservable transitions in the LPN and m denotes that of places in the LPN. Hence, the on-line computational cost of the proposed algorithm increases with the number of observed events. However, in practice, our experience shows that in the examined cases, compared with those presented in the literature, an optimal solution is obtained in a short time by solving the ILPPs on a PC equipped with a standard solver of optimization tool.

3.5 Example

This subsection provides some experimental results of the algorithm proposed in this paper. The obtained computational time refers to the CPU seconds of a notebook computer under the Windows 7 operating system with Intel CPU Core 2.6 GHz, 8 GB memory and a standard optimization solver.

In order to show the advantage and efficiency of the proposed on-line algorithm, let us consider a large example shown in Fig. 2, which is taken from [9]. This example is an LPN that models a manufacturing system. The LPN is composed by 46 places, 39 transitions and the event set is $E = \{a, b, c, d, e, g, l\}$. The set of observable transitions T_o consists of transitions from t_1 to t_{13} such that $T(a) = \{t_1\}$, $T(b) = \{t_2, t_3, t_{11}\}$, $T(c) = \{t_4\}$, $T(d) = \{t_5, t_{10}, t_{13}\}$, $T(e) = \{t_6\}$, $T(g) = \{t_7, t_8, t_{12}\}$ and $T(l) = \{t_9\}$. The secret is defined by the following set: $S = \{M \in \mathbb{N}^{46} | M(p_{16}) + M(p_{19}) + M(p_{26}) + M(p_{27}) \leq 0\}$. By applying the method in [31] to this example, the system is not current-state opaque wrt the secret. However, it takes more than 1800 seconds to obtain this result due to the computation of the BRG and its observer.

Now, we apply Algorithm 1 to this example. In particular, the performance of Algorithm 1 is presented in Table 1, where the first column represents the evolution of the system, N_{var} and N_{con} indicate the numbers of variables and constraints of ILPP 1, respectively. The fourth column shows the CPU time in seconds for solving ILPP 1 and the fifth column is the output of Algorithm 1 at each step. From Table 1, we can see that the observed event sequence $w = aegl$ is not current-state opaque wrt the secret S. Hence, according to Definition 4, we conclude that the LPN system is not current-state opaque wrt the secret.

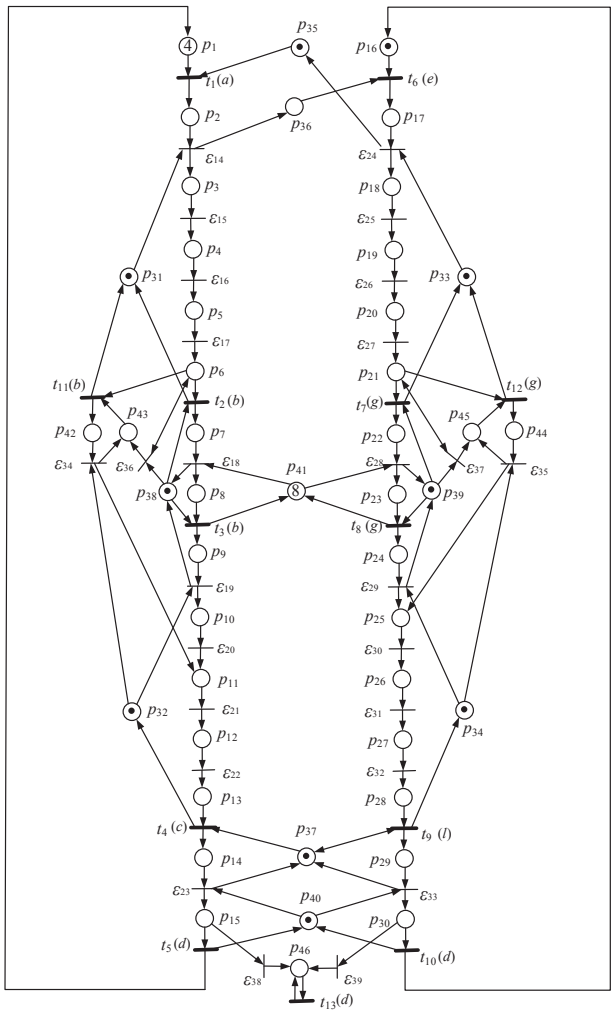

Fig. 2. LPN modeling a manufacturing system [9].

Table 1
Performance of Algorithm 1 on the net in Fig. 2

Action	N_{var}	N_{con}	time/(s)	$\Phi(w)$
no observable event occurs	72	92	9×10^{-3}	{Y}
observable event a occurs	111	140	1.2×10^{-2}	{Y}
observable event e occurs	150	188	1.4×10^{-2}	{Y}
observable event g occurs	189	236	1.6×10^{-2}	{Y}
observable event l occurs	228	284	1.7×10^{-2}	{N}

4 Verifying current-state opacity in a decentralized approach

In this section, we extend the study of on-line verification of current-state opacity to a decentralized architecture where a set of local intruders observes the system and each intruder can observe only a part of the observable events. Each local intruder has the full knowledge of the system model and communicates with a coordinator to

infer the secret. More precisely, each local intruder observes the system by its own observation mask, checks the opacity property for the local observation, and then reports the result to the coordinator. A protocol is proposed for the communication between the local intruders and the coordinator.

4.1 The decentralized architecture

In the following, we introduce the decentralized architecture that is used in this paper. This architecture is presented in [12] and [33] in the framework of automata, and also adopted [10] in the framework of PNs. The decentralized architecture for verifying current-state opacity is shown in Fig. 3, where a set $\mathcal{J} = \{1, 2, \ldots, J\}$ of local intruders works to verify if the behaviour of the system remains in the secret under the given observation. Different local intruders can have different observation masks. Hence, we define $T_{o,j} \subseteq T_o$ as the set of locally observable transitions for each local intruder $j \in \mathcal{J}$. Any observable transition is observed by at least one local intruder, i.e., $\bigcup_{j\in\mathcal{J}} T_{o,j} = T_o$. The set of locally unobservable transitions is defined as $T_{u,j} = T \setminus T_{o,j}$. In the following, $n_{o,j}$ and $n_{u,j}$ denote the numbers of the locally observable and unobservable transitions, respectively.

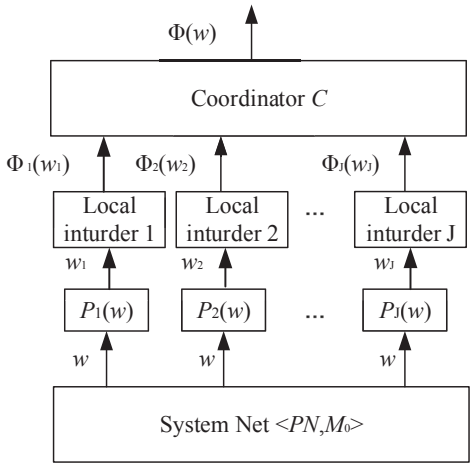

Fig. 3. The decentralized architecture.

Now, we define the labeling function associated with the jth local intruder as follows:

$$\lambda_j(t) = \begin{cases} \lambda(t), \ if \ \ t \in T_{o,j} \\ \varepsilon, \quad otherwise. \end{cases} \tag{8}$$

Moreover, for each $j \in \mathcal{J}$, $E_j \subseteq E$ denotes the set of observable labels by the jth local intruder, and the projection over E_j, for $j \in \mathcal{J}$, $P_j : E^* \to E_j^*$ is defined as follows: for all $w \in E^*$ and $e \in E$, 1) $P_j(\varepsilon) = \varepsilon$ and 2) $P_j(we) = P_j(w)e$ if $e \in E_j$ and $P_j(we) = P_j(w)$ otherwise.

As shown in Fig. 3, each local intruder $j \in \mathcal{J}$ checks the opacity property for its own observation $w_j = P_j(w)$, and depending on its output $\Phi_j(w_j)$, it transmits this result to a coordinator by a given protocol. The coordinator analyses the information from the local intruders and then generates a global output $\Phi(w)$.

The following assumptions are introduced for the decentralized architecture.

- A2) The $T_{u,j}$-induced subnet $PN_{u,j}\angle_{T_{u,j}}PN$ and the $T_{o,j}$-induced subnet $PN_{o,j}\angle_{T_{o,j}}PN$ are acyclic for any local intruder $j \in \mathcal{J}$.
- A3) For each label $e \in E$, there exists at least one local intruder that can observe all transitions whose label is e.
- A4) The output sent by each local intruder is received by the coordinator correctly.
- A5) There is no delay in the communication between the local intruders and the coordinator.

4.2 The local intruder specification

First, we appropriately modify the formulation of ILPP 1 of Algorithm 1 proposed in Section 3 for each local intruder $j \in \mathcal{J}$.

Let w_j be an observation of the jth local intruder, and w_j be associated with the firing sequence $\sigma_j \in T^*$ such that $w_j = \lambda_j(\sigma_j)$ by using the extended form of the labeling function $\lambda_j : T^* \to E_j^*$. We define $\mathcal{S}_j(w_j) = \{\sigma_j \in L(PN, M_0) | \lambda_j(\sigma_j) = w_j\}$ as the set of firing sequences consistent with w_j for the jth local intruder and $\mathcal{C}_j(w_j) = \{M \in \mathbb{N}^m | \exists \sigma_j \in \mathcal{S}(w_j) : M_0[\sigma\rangle M\}$ as the set of markings consistent with w_j for the jth local intruder.

Moreover, we denote $C_u^j = Post_u^j - Pre_u^j$ and $C_o^j = Post_o^j - Pre_o^j$ as the restrictions of the incidence matrix $C = Post - Pre$ to $T_{u,j}$ and $T_{o,j}$, respectively. The following proposition is immediately obtained from Proposition 1 in Section 3 to provide a local linear algebraic representation of a sequence $\sigma_j \in T^*$ that is consistent with w_j.

Proposition 3 *Consider an LPN system $G = (PN, M_0, E, \lambda)$ with language $\mathcal{L}(PN, M_0)$ and the jth local intruder satisfying assumption A2. Given an observation $w_j \in \mathcal{L}(PN, M_0)$ denoted by $w_j = e_1^j e_2^j \ldots e_d^j$ ($e_i^j \in E$ for $i = 1, 2, \ldots, d$ denotes the ith locally observed event), there exists at least one sequence $\sigma_j = \sigma_{u_1}^j \sigma_{o_1}^j \sigma_{u_2}^j \sigma_{o_2}^j \ldots \sigma_{u_d}^j \sigma_{o_d}^j \sigma_{o_{d+1}}^j$ with $|\sigma_{u_i}| \geq 0$ for $i = 1, 2, \ldots, d$ enabled at the initial marking M_0 such that $\lambda_j(\sigma_j) = w_j = e_1^j e_2^j \ldots e_d^j$ iff there exist $2d+1$ firing vectors $\sigma_{u_1}^j, \sigma_{u_2}^j, \ldots, \sigma_{u_{d+1}}^j, \sigma_{o_1}^j, \sigma_{o_2}^j, \ldots, \sigma_{o_d}^j$ that satisfy the following set of constraints denoted by $\rho_j(M_0, w_j)$:*

$$
\begin{cases}
\vec{\sigma}_{u_i^j} \in \mathbb{N}^{n_{u,j}}, \quad for\ i = 1, \ldots, d+1 & (a)\\
\vec{\sigma}_{o_i^j} \in \mathbb{N}^{n_{o,j}}, \quad for\ i = 1, \ldots, d & (b)\\
C_u^j \sum\limits_{i=1}^{v} \vec{\sigma}_{u_i^j} \geq Pre_o^j \cdot \vec{\sigma}_{o_v}^j - M_0 - C_o^j \sum\limits_{i=1}^{v-1} \vec{\sigma}_{o_i}^j, \\
for\ v = 1, \ldots d & (c)\\
M_0 + C_u^j \sum\limits_{i=1}^{d+1} \vec{\sigma}_{u_i}^j + C_o^j \sum\limits_{i=1}^{d} \vec{\sigma}_{o_i}^j \geq \vec{0} & (d)\\
\sum_{t_k \in T(e_1^j)} \vec{\sigma}_{o_1}^j(t_k) = 1\\
\sum_{t_k \in T(e_2^j)} \vec{\sigma}_{o_2}^j(t_k) = 1\\
\ldots\\
\sum_{t_k \in T(e_d^j)} \vec{\sigma}_{o_d}^j(t_k) = 1 & (e)\\
\sum_{t_k \notin T(e_1^j)} \vec{\sigma}_{o_1}^j(t_k) = 0\\
\sum_{t_k \notin T(e_2^j)} \vec{\sigma}_{o_2}^j(t_k) = 0\\
\ldots\\
\sum_{t_k \notin T(e_d^j)} \vec{\sigma}_{o_d}^j(t_k) = 0.
\end{cases} \tag{9}
$$

Similarly to Remark 1, when $w = \varepsilon$, the set of constraints $\rho_j(M_0, w_j)$ can be rewritten as follows:

$$
\begin{cases}
\vec{\sigma}_{u_1^j} \in \mathbb{N}^{n_{u,j}},\\
M_0 + C_u^j \cdot \vec{\sigma}_{u_1}^j \geq \vec{0}.
\end{cases} \tag{10}
$$

Accordingly, we can obtain the formulation of ILPP 1 for each local intruder $j \in \mathcal{J}$ as follows:

$$
\begin{cases}
z_q = max & x_q^T \cdot M\\
s.t. & \rho_j(M_0, w_j)\\
& M = M_0 + C_u^j \sum\limits_{i=1}^{d+1} \vec{\sigma}_{u_i}^j + C_o^j \sum\limits_{i=1}^{d} \vec{\sigma}_{o_i}^j.
\end{cases} \tag{11}
$$

4.3 Main results for the decentralized approach

In the following, we present some results that can provide a rule for the coordinator to produce a global output about the verification of current-state opacity for a given observation based on the information of the local intruders.

Definition 6 *Assume that $w = e_1 e_2 \ldots e_h$ $(h \geq 1)$ is an observed word and $w_j = e_1^j e_2^j \ldots e_d^j$ $(d \leq h)$ is the word projection of the local intruder $j \in \mathcal{J}$. $\mathcal{J}^*(w) = \{j \in \mathcal{J} | e_h = e_d^j\}$ is defined as the set of local intruders that can observe the last event of w.*

Remark 3 *By assumption A3, the set of local intruders that can observe the last event of w is not empty, i.e., $\mathcal{J}^*(w) \neq \emptyset$. In particular, if $w = \varepsilon$, then $\mathcal{J}^*(w) = \mathcal{J}$ always holds.*

Proposition 4 shows the relations between the set of sequences consistent with w_j and the set of sequences consistent with w.

Proposition 4 *Consider an observation $w \in \mathcal{L}(PN, M_0)$. Under assumption A3, it holds $\mathcal{S}(w) \subseteq \mathcal{S}_j(w_j)$ for each local intruder $j \in \mathcal{J}^*(w)$. .*

Proof: By Remark 3 (where assumption A3 necessarily holds), we have $\mathcal{J}^*(w) \neq \emptyset$. Since $T_{o,j} \subseteq T_o$ and $E_j \subseteq E$ for each $j \in \mathcal{J}^*(w)$, we infer $w_j \in w$. If there exists $\sigma \in \mathcal{S}(w)$ with $w = \lambda(\sigma)$, we can find a sequence $w_j \in w$ such that $w_j = \lambda_j(\sigma)$. Hence, σ is a sequence consistent with w and also a sequence consistent with w_j. We conclude that $\mathcal{S}(w) \subseteq \mathcal{S}_j(w_j)$ holds. □

Based on Proposition 4, Corollary 2 shows the relations between the set of markings consistent with w_j and the set of markings consistent with w.

Corollary 2 *Consider an observation $w \in \mathcal{L}(PN, M_0)$. Under assumption A3, it holds $\mathcal{C}(w) \subseteq \mathcal{C}_j(w_j)$ for each local intruder $j \in \mathcal{J}^*(w)$.*

Proof: It follows immediately from Proposition 4 and the state equation. □

The following proposition provides a rule for a coordinator to determine whether a given observation of the system is not current-state opaque according to the information of a local intruder.

Proposition 5 *Let $G = (PN, M_0, E, \lambda)$ be an LPN system and S be a secret. If there exists a local intruder $j \in \mathcal{J}^*(w)$ that provides $\Phi_j(w_j) = \{N\}$, then it holds $\Phi(w) = \{N\}$.*

Proof: If there exists a local intruder $j \in \mathcal{J}^*(w)$ that provides $\Phi_j(w_j) = \{N\}$, then by Definitions 3 and 5, we infer $\mathcal{C}_j(w_j) \subseteq S$. According to Corollary 2, $\mathcal{C}(w) \subseteq \mathcal{C}_j(w_j)$ holds and it implies that $\mathcal{C}(w) \subseteq S$. Thus, $\Phi(w) = \{N\}$ holds. □

Corollary 3 *Let $G = (PN, M_0, E, \lambda)$ be an LPN system and S be a secret. If the system is not current-state opaque wrt S for any local intruder $j \in \mathcal{J}^*(w)$, then the system is not current-state opaque wrt S.*

Proof: It follows immediately from Definition 4 and Proposition 5. □

Algorithm 2. Protocol 1
Input: $G = (PN, M_0, E, \lambda)$, S
Output: $\Phi(w)$
Step 1. *Initializing the variables of the algorithm*
$w := \varepsilon$, $w_j := \varepsilon$, $\Phi(w) := \emptyset$
Step 2. *Performance of the local intruders*
for each local intruder $j \in \mathcal{J}^*(w)$ **do**
computes $\Phi_j(w_j)$ by Algorithm 1.
if $\Phi_j(w_j) = \{N\}$ **then**
local intruder j transmits $\Phi_j(w_j)$ to the coordinator
end if
end for
Step 3. *Determining the output of the coordinator*
If the coordinator receives $\Phi_j(w_j) = \{N\}$ **then**
it sets $\Phi(w) = \{N\}$ **go to Step 5**
Step 4. *Recording the events*
Wait until an event $e \in E$ occurs
$w := we$, $w_j := w_j P_j(e)$, $\Phi(w) := \emptyset$
go to Step 2
Step 5. End

4.4 The decentralized algorithm for the current-state opacity

In this subsection, we present Algorithm 2 that the local intruders and the coordinator have to apply in order to verify the current-state opacity. Step 1 initializes the variables of Algorithm 2. In Step 2, each local intruder $j \in \mathcal{J}^*(w)$ verifies the current-state opacity of the system on the basis of its own observation. If $\Phi_j(w_j) = \{N\}$, then j transmits its result to the coordinator. Step 3 is performed by the coordinator: according to Proposition 5, if there exists a local intruder $j \in \mathcal{J}^*(w)$ that provides $\Phi_j(w_j) = \{N\}$, then $\Phi(w) = \{N\}$ holds. In this case, by Corollary 3, the system is not current-state opaque wrt S. Moreover, Step 4 updates the observation by waiting a new event $e \in E$ occurrence.

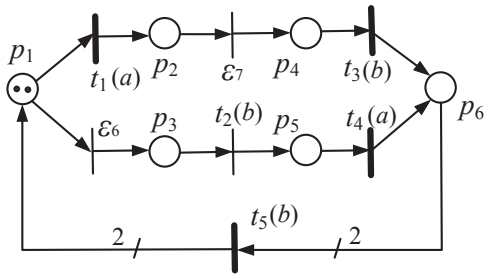

Fig. 4. An LPN system considered in Example 2.

Example 2 *Let us consider the LPN in Fig. 4 with two local intruders* $\mathcal{J} = \{1, 2\}$ *whose sets of observable transitions are* $T_{o,1} = \{t_1, t_4\}$ *and* $T_{o,2} = \{t_2, t_3, t_5\}$*, respectively. The system events are* $E = \{a, b\}$ *with* $E_1 = \{a\}$ *and* $E_2 = \{b\}$*. The initial marking is* $M_0 = [2, 0, 0, 0, 0, 0]^T$ *and the secret is* $S = \{M \in \mathbb{N}^6 | X \cdot M \leq K\}$ *with* $X = [0, 1, 1, 0, 0, 0]$ *and* $K = 1$.

Suppose that no observable event occurs in the system, i.e., $w = \varepsilon$, $w_1 = \varepsilon$ *and* $w_2 = \varepsilon$*. Local intruders 1 and 2 provide* $\Phi_1(w_1) = \{Y\}$ *and* $\Phi_2(w_2) = \{Y\}$*, respectively. According to Algorithm 2, no local intruder transmits its result to the coordinator. Consequently, the coordinator remains silent.*

Suppose that the first observable event a occurs in the system, hence $w = a$, $w_1 = a$ *and* $w_2 = \varepsilon$*. Since* $\mathcal{J}^*(w) = \{1\}$*, local intruder 1 provides* $\Phi_1(w_1) = \{Y\}$*. According to Algorithm 2, no local intruder transmits its result to the coordinator and also in this case the coordinator remains silent.*

Finally, assume that the second observable event b occurs: $w = ab$, $w_1 = a$ *and* $w_2 = b$*. Since* $\mathcal{J}^*(w) = \{2\}$*, local intruder 2 provides* $\Phi_2(w_2) = \{N\}$*. According to Algorithm 2, this intruder transmits its result to the coordinator. Then, the coordinator sets* $\Phi(w) = \{N\}$ *and according to Definition 4, the system is not current-state opaque wrt* S*.*

5 Conclusion and future work

In this paper, we propose an on-line intruder in the labeled Petri Net (LPN) framework. The intruder observes and stores the event sequence of the LPN system and decides on-line whether the given observation of the system is current-state opaque or not. To this aim, an Integer Linear Programming problems is defined and we prove that, on the basis of the provided solutions at each observed event, it is possible to decide if the system is not current-state opaque. Moreover, in order to deal with decentralized system settings, the proposed on-line approach is extended to a decentralized architecture where a set of local intruders communicate with a coordinator.

Compared with the existing approach of [31], we enlighten that the proposed methodology for opacity verification avoids the enumeration of the PN markings and can be applied also to unbounded nets. Moreover, the on-line approach allows us to avoid the redesign of the intruder when the system changes because it is only necessary to give the new LPN structure to the algorithm. By applying the proposed algorithm to a large example, we show its efficiency.

Future works will focus on improving the decentralized approach by obtaining results also in the case that the system observation is current-state opaque wrt the secret for the local intruders.

References

[1] Badouel, E., Bednarczyk, M., Borzyszkowski, A., Caillaud, B., & Darondeau, P. (2007). Concurren-

t secrets. *Discrete Event Dynamic Systems*, 17(4), 425–446.

[2] Bai, L. P., Wu, N. Q., Li, Z. W., & Zhou, M. C. (2016). Optimal one-wafer cyclic scheduling and buffer space configuration for single-arm multicluster tools with linear topology, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 46(10), 1456–1467.

[3] Basile, F., Chiacchio, P., & Tommasi, G. De. (2009). An efficient approach for online diagnosis of discrete event systems. *IEEE Transactions on Automatic Control*, 54(4), 748–759.

[4] Basile, F., Chiacchio, P., & Tommasi, G. De. (2012). On K-diagnosability of Petri nets via integer linear programming. *Automatica*, 48, 2047–2058.

[5] Bryans, J. W., Koutny, M., & Ryan, P. Y. (2005). Modelling opacity using Petri nets. *Electronic Notes in Theoretical Computer Science*, 121, 101–115.

[6] Bryans, J. W., Koutny, M., Mazaŕe, L., & Ryan, P. Y. (2008). Opacity generalised to transition systems. *International Journal of Information Security*, 7(6),421–435.

[7] Busi, N., & Gorrieri, R. (2004). A survey on noninterference with Petri nets. *Lectures on Concurrency and Petri Nets*, Springer, 328–344.

[8] Cabasino, M. P., Giua, A., & Seatzu, C. (2010). Fault detection for discrete event systems using Petri nets with unobservable transitions. *Automatica*, 46(9), 1531–1539.

[9] Cabasino, M. P., Giua, A., Pocci, M., & Seatzu, C. (2011). Discrete event diagnosis using labeled Petri nets. An application to manufacturing systems. *Control Engineering Practice*, 19(9), 989–1001.

[10] Cabasino, M. P., Giua, A., Poali, A., & Seatzu, C. (2013). Decentralized diagnosis of discrete-event systems using labeled Petri nets. *IEEE Transactions on Systems Man and Cybernetics: Systems*, 43(6), 1477–1485.

[11] Corona, D., Giua, A., & Seatzu, C. (2004). Marking estimation of Petri nets with silent transitions. *Proceedings of the 43rd IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas (pp. 966–971).

[12] Debouk, R., Lafortune, S., & Teneketzis, D. (2000). Coordinated decentralized protocols for failure diagnosis of discrete-event systems. *Discrete Event Dynamic Systems*, 10(1), 33–86.

[13] Dotoli, M., Fanti, M. P., Mangini, A. M., & Ukovich, W. (2009). On-line fault detection in discrete event systems by Petri nets and integer linear programming. *Automatica*, 45, 2665–2672.

[14] Fanti, M. P., Mangini, A. M., & Ukovich, W. (2013). Fault detection by labeled Petri nets in centralized and disributed approaches. *IEEE Transactions on Automation Science and Engineering*, 10(2), 392–404.

[15] Giua, A., DiCesare, F., & Silva, M. (1992). Generalized mutual exclusion contraints on nets with uncontrollable transitions. *Proceedings of the 1992 IEEE International Conference on Systems, Man and Cybernetics*, Chicago, IL, USA (pp. 974–979).

[16] Giua, A., & Seatzu, C. (2005). Fault detection for discrete event systems using Petri nets with unobservable transitions. *Proceedings of the 44th IEEE Conference on Decision and Control, and the 2005 European Control Conference*, Seville, Spain (pp. 6323–6328).

[17] Hadj-Alouane, N. B., Lafrance, S., Lin, F., Mullins, J., & Yeddes, M. M. (2005). On the verification of intransitive noninterference in mulitlevel security. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 35(5), 948–958.

[18] Li, Z. W., & Zhao, M. (2008). On controllability of dependent siphons for deadlock prevention in generalized Petri nets. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 38(2), 369–384.

[19] Li, Z. W., Liu, G. Y., Hanisch, M-H., & Zhou, M. C. (2012). Deadlock prevention based on structure reuse of Petri net supervisors for flexible manufacturing systems, *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 42(1), 178–191.

[20] Ma, Z. Y., Li, Z. W., & Giua, A. (2015). Design of optimal Petri net controllers for disjunctive generalized mutual exclusion constraints, *IEEE Transactions on Automatic Control*, 60(7), 1774–1785.

[21] Paoli A., & Lin, F. (2012). Decentralized opacity of discrete event systems. *Proceedings of the 2012 American Control Conference*, Montréal, Canada (pp. 6083–6088).

[22] Peterson, J. L. (1981). *Petri net theory and the modeling of systems*. Englewood Cliffs, NJ, USA: Prentice Hall.

[23] Reiter, M. K., & Rubin, A. D. (1998). Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security*, 1(1), 66–92.

[24] Saboori, A., & Hadjicostis, C. N. Notions of security and opacity in discrete event systems. (2007). *Proceedings of the 46th IEEE Conference on Decision and Control*, New Orleans, Louisiana, USA (pp. 5056–5061).

[25] Saboori, A., & Hadjicostis, C. N. Verification of initial-state opacity in security applications of DES. (2008). *Proceedings of the 9th International Workshop on Discrete Event Systems*, Goteborg, Sweden (pp. 328–333).

[26] Saboori, A., & Hadjicostis, C. N. (2011). Verification of k-step opacity and analysis of its complexity. *IEEE Transactions on Automation Science and Engineering*, 8(3), 549–559.

[27] Saboori, A., & Hadjicostis, C. N. (2013). Verification of initial-state opacity in security applications of discrete event systems. *Information Sciences*, 246, 115–132.

[28] Saboori, A., & Hadjicostis, C. N. (2014). Current-state opacity formulations in probabilistic finite automata. *IEEE Transactions on Automatic Control*,

59(1), 120–133.

[29] Shmatikov, V. (2004). Probabilistic analysis of an anonymity system. *Journal of Computer Security*, 12(3), 355–377.

[30] Tong, Y., Li, Z. W., Seatzu, C., & Giua, A. (2015a). Verification of initial-state opacity in Petri nets. *Proceedings of the 54th IEEE Conference on Decision and Control*, Osaka, Japan (pp. 344–349).

[31] Tong, Y., Li, Z. W., Seatzu, C., & Giua, A. (2015b). Verification of current-state opacity using Petri nets. *Proceedings of the 2015 American Control Conference*, Chicago, IL, USA (pp. 1935–1940).

[32] Uzam, M., Li, Z. W., Gelen, G., & Zakariyya, R. S. (2016). A divide-and-conquer-method for the synthesis of liveness enforcing supervisors for flexible manufacturing systems, *Journal of Intelligent Manufacturing*, 27(5), 1111-1129.

[33] Wu, Y., & Lafortune, S. (2013). Comparative analysis of related notions of opacity in centralized and coordinated architectures. *Discrete Event Dynamic Systems*, 23(3), 307–339.