Verification of Initial-State Opacity in Petri Nets

Yin Tong[1], Zhiwu Li[2], Carla Seatzu[3] and Alessandro Giua[4]

Abstract

A Petri net system is said to be initial-state opaque if its initial state remains opaque to an external observer (called an intruder). In other words, the intruder is never able to ascertain that the initial state belongs to a given set of states (called a secret) based on its observation of the system's evolution. This paper addresses the problem of verifying initial-state opacity in discrete event systems (DES) modeled by labeled Petri nets. An efficient approach to verifying initial-state opacity is proposed based on the notion of *basis reachability graph* (BRG).

To appear as:

Y. Tong, Z.W. Li, C. Seatzu, A. Giua, "Verification of Initial-State Opacity in Petri Nets", 54nd IEEE Conf. on Decision and Control (Osaka, Japan), Dec. 15-18, 2015.

---

[1]Yin Tong is with the School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China, and also with DIEE, University of Cagliari, 09123 Cagliari, Italy yintong@stu.xidian.edu.cn

[2]Zhiwu Li is with the Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau, Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia, and also with the School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China zhwli@xidian.edu.cn

[3]Carla Seatzu is with the Department of Electrical and Electronic Engineering, University of Cagliari, 09123 Cagliari, Italy seatzu@diee.unica.it

[4]Alessandro Giua is with Aix Marseille Université, CNRS, ENSAM, Université de Toulon, LSIS UMR 7296, Marseille 13397, France and also with DIEE, University of Cagliari, Cagliari 09123, Italy alessandro.giua@lsis.org; giua@diee.unica.it

I. INTRODUCTION

Opacity is an information flow property [1], [2], [3], [4] which relates to the system's ability to hide a secret behavior from an intruder [5], [6], [7], [8], [9]. In DES models, the secret is usually defined as a subset of the state space or a language (subsequent opacity properties are referred to as *state-based opacity* and *language-based opacity*, respectively), and the intruder is modeled as an observer that has full knowledge of the system's structure but only has partial observability over the system's evolution. Based on its observation, the intruder tries to infer the secret.

Initial-state opacity is a state-based opacity property. A system is said to be initial-state opaque if, given a set of secret states, by observing the sequence of events generated by the system, the intruder will never be able to infer that the system's evolution started from one of the secret states.

In recent years, initial-state opacity has been extensively studied in the framework of automata [8], [10], [11]. It has been proved that the verification of initial-state opacity is PSPACE-complete [11]. Saboori and Hadjicostis [10], [11] have shown that by constructing the *initial-state estimator* for a given nondeterministic finite automaton (NFA), initial-state opacity can be verified with complexity $\mathcal{O}(2^{|X|^2})$, where X is the set of states of the automaton. An initial-state estimator is a deterministic finite automaton (DFA) whose states denote the set of initial states where an observed word could have started and the current states that it yields. As long as an initial-state estimator is built, there is no need to reconstruct it when the secret is modified. For a specific secret, *verifiers* are introduced in [11] to study initial-state opacity. Instead of precisely estimating the initial state, the verifier only records if a state is reachable from secret/non-secret states. Therefore, the verification complexity is reduced to $\mathcal{O}(4^{|X|})$. Meanwhile, Wu and Lafortune [8] propose a more efficient method whose complexity is $\mathcal{O}(2^{|X|})$. They show that the observer of the reverse automaton can be used to estimate the initial state.

Bryans et al. [4] proved that the verification of initial-state opacity for bounded PNs is decidable when the initial state is defined as a finite set of initial markings and the secret is a subset of it. However, in PNs the initial-state opacity problem is very difficult in general, and so far no efficient method has been proposed yet. For bounded PNs we may construct its reachability graph (RG), which is an automaton, so that the aforementioned approaches in automata could be applied. Nevertheless, this approach will inevitably suffer from the state explosion problem.

As a compact description of the RG, *basis reachability graph* (BRG) has been used to solve problems of state estimation, fault diagnosis [12] and current-state opacity [13]. The advantage of this technique is that only part of the reachability space, i.e., the set of *basis markings* (see Section IV.A), has to be enumerated, and all other reachable markings can be characterized in terms of linear algebra. The BRG of a PN, in general, is smaller than the RG, but it well characterizes both the reachable markings and the behavior (language) of the corresponding PN.

In this paper, the verification of initial-state opacity in bounded *labeled Petri nets* is addressed. The secret is defined as a subset of the reachable markings. A labeled Petri net is initial-state opaque with respect to a secret if the intruder can never infer that the observed sequence origins from a secret marking. It is known that a net is initial-state opaque if and only if the language generated from secret markings is a subset of the language generated

from non-secret markings. Therefore, the initial-state opacity problem in bounded labeled Petri nets is transformed into the language containment problem in its RG. Considering that the intruder would never distinguish a possible non-secret marking that is reachable from a secret initial marking by firing only unobservable transitions, we make the following reasonable assumption: *all markings reachable from a secret marking by firing only unobservable transitions belong to the secret*. Under this assumption, we show that initial-state opacity of a bounded net can be verified by justifying the language containment in the corresponding BRG. Therefore, compared with using RG, the approach proposed in this work is more efficient in general.

II. PRELIMINARIES

A. Automata

A *non-deterministic finite-state automaton* (NFA) is a 4-tuple $\mathcal{A} = (X, E, \Delta, x_0)$, where X is the finite *set of states*, $E = \{a, b, \cdots\}$ is the *alphabet*, $\Delta \subseteq X \times E_\varepsilon \times X$ is the *transition relation* with $E_\varepsilon = E \cup \{\varepsilon\}$, where ε is the empty word describing unobservable events, and $x_0 \in X$ is the *initial state*. The transition relation specifies the dynamics of the NFA: if $(x, e, x') \in \Delta$, then from state x the occurrence of event $e \in E_\varepsilon$ yields state x'. The transition relation can be extended to $\Delta^* \subseteq X \times E^* \times X$: $(x_{j0}, w, x_{jk}) \in \Delta^*$ if there exists a sequence of events and states $x_{j0}e_{j1}x_{j1}\cdots x_{jk-1}e_{jk}x_{jk}$ such that $\sigma = e_{j1}\dots e_{jk}$ generates the word $w \in E^*$, $x_{ji} \in X$ for $i = 0, 1, \dots, k$, and $e_{ji} \in E_\varepsilon$, $(x_{ji-1}, e_{ji}, x_{ji}) \in \Delta$ for $i = 1, 2, \dots, k$. An NFA is denoted as $\mathcal{A} = (X, E, \Delta)$ in the case where the initial state could be any state from X.

The *generated language* of an automaton $\mathcal{A} = (X, E, \Delta)$ from a state $x \in X$ is defined as

$$\mathcal{L}(\mathcal{A}, x) = \{w \in E^* | \exists x' \in X : (x, w, x') \in \Delta^*\}.$$

Generally, given a set of states $Y \subseteq X$, we define $\mathcal{L}(\mathcal{A}, Y) = \bigcup_{x \in Y} \mathcal{L}(\mathcal{A}, x)$ the language generated from the states in Y.

B. Petri nets

A *Petri net* is a structure $N = (P, T, Pre, Post)$, 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 functions* that specify the arcs directed from places to transitions, and vice versa. The incidence matrix of a net is denoted by $C = Post - Pre$.

A *marking* is a vector $M : P \to \mathbb{N}$ that assigns to each place a non-negative integer number of tokens, represented by black dots. The marking of place p is denoted by $M(p)$. For economy of space, markings can also be denoted as $M = \sum_{p \in P} M(p) \cdot p$ (see Fig.5). A *Petri net system* $\langle N, M_0 \rangle$ is a net N with *initial marking* M_0.

A transition t is *enabled* at marking M if $M \geq Pre(\cdot, t)$ and may fire yielding a new marking $M' = M + C(\cdot, t)$. We write $M[\sigma\rangle$ to denote that the sequence of transitions $\sigma = t_{j1} \cdots t_{jk}$ is enabled at M, and $M[\sigma\rangle M'$ to denote that the firing of σ yields M'. Given a sequence $\sigma \in T^*$, we call $\pi : T^* \to \mathbb{N}^n$ the function that associates with σ the Parikh vector $y = \pi(\sigma) \in \mathbb{N}^n$, i.e., $y(t) = k$ if transition t appears k times in σ.

A marking M is *reachable* in $\langle N, M_0\rangle$ if there exists a sequence σ such that $M_0[\sigma\rangle M$. The set of all markings reachable from M_0 defines the *reachability set* of $\langle N, M_0\rangle$ and is denoted by $R(N, M_0)$. A PN system is *bounded* if there exists a non-negative integer $k \in \mathbb{N}$ such that for any place $p \in P$ and for any reachable marking $M \in R(N, M_0)$, $M(p) \leq k$ holds.

A *labeled Petri net* (LPN) is a 4-tuple $G = (N, M_0, E, \ell)$, where $\langle N, M_0\rangle$ is the PN system, E is the *alphabet* (a set of labels) and $\ell : T \to E \cup \{\varepsilon\}$ is the *labeling function* that assigns to each transition $t \in T$ either a symbol from E or the empty word ε. Therefore, the set of transitions can be partitioned into two disjoint sets $T = T_o \cup T_u$, where $T_o = \{t \in T | \ell(t) \in E\}$ is the set of observable transitions and $T_u = \{t \in T | \ell(t) = \varepsilon\}$ is the set of unobservable transitions. The labeling function can be extended to firing sequences $\ell : T^* \to E^*$, i.e., $\ell(\sigma t) = \ell(\sigma)\ell(t)$ with $\sigma \in T^*$ and $t \in T$. The *unobservable reach* of a marking M is defined as $\mathcal{U}(M) = \{M' \in \mathbb{N}^n | \exists \sigma_u \in T_u^* : M[\sigma_u\rangle M'\}$, i.e., the set of markings reachable from M by firing unobservable transitions.

Given an LPN $G = (N, M_0, E, \ell)$ and a marking $M \in R(N, M_0)$, we define the language generated from M as $\mathcal{L}(N, M) = \{w \in E^* | \exists \sigma \in T^* : M[\sigma\rangle$ and $\ell(\sigma) = w\}$. The *generated language* of G is $\mathcal{L}(N, M_0)$. Furthermore, given a set of markings $Y \subseteq R(N, M_0)$ of G, we define $\mathcal{L}(N, Y) = \bigcup_{M \in Y} \mathcal{L}(N, M)$ the language generated from markings in Y.

Given an LPN $G = (N, M_0, E, \ell)$ and the set of unobservable transitions T_u, the *T_u-induced subnet* $N' = (P, T', Pre', Post')$ of N, is the net that removes all observable transitions in T_o, where Pre' and $Post'$ are the restriction of Pre, $Post$ to T_u. The incidence matrix of the T_u-induced subnet is denoted by $C_u = Post' - Pre'$.

III. Initial-State Opacity in Petri Nets

A. Initial-state opacity

Definition 3.1: Given an LPN $G = (N, M_0, E, \ell)$, a *secret* is a set of reachable markings $S \subseteq R(N, M_0)$. A marking $M \in S$ is said to be a *secret marking*. Markings in $\overline{S} = R(N, M_0) \setminus S$ are *non-secret markings*. $\diamond$

Definition 3.2: Let $G = (N, M_0, E, \ell)$ be an LPN and $S \subseteq R(N, M_0)$ be a secret. G is said to be initial-state opaque wrt S if

$$\forall M \in S,\ \forall w \in \mathcal{L}(N, M)\ \exists M' \in \overline{S} : w \in \mathcal{L}(N, M'). \qquad \diamond$$

In simple words, a PN is initial-state opaque if for any word w that can be observed starting from some secret markings in S, there always exists (at least) one non-secret marking from which w could also be generated so that the intruder cannot establish if the system started its evolution from a secret or a non-secret marking.

B. Verification of initial-state opacity using RG

Based on the given secret, we define the *secret language* and the *non-secret language*.

Definition 3.3: Given an LPN $G = (N, M_0, E, \ell)$ and a secret $S \subseteq R(N, M_0)$, its *secret language* is defined as

$$\mathcal{L}(N, S) = \bigcup_{M \in S} \mathcal{L}(N, M),$$

and its *non-secret language* is defined as

$$\mathcal{L}(N, \overline{S}) = \bigcup_{M \in \overline{S}} \mathcal{L}(N, M).$$

◇

Lemma 3.4: Let $G = (N, M_0, E, \ell)$ be an LPN and S be a secret. G is initial-state opaque wrt S if and only if $\mathcal{L}(N, S) \subseteq \mathcal{L}(N, \overline{S})$. ◇

Proof: Follows from Definitions 3.2 and 3.3. ■

Lemma 3.4 shows that an LPN is opaque with respect to a given secret if and only if its secret language is a subset of the non-secret language. As a result, the initial-state opacity problem in PNs is equivalent to the language containment problem. Therefore, in the case of bounded nets, by constructing the RG, all methods of verifying language containment in automata can be applied to solving the opacity problem. The complexity of checking language containment of two NFA having the same number of states is $\mathcal{O}(4^{|X|})$, where X is the set of states [14]. Therefore, the size of the RG greatly affects the efficiency of verifying initial-state opacity in bounded PNs.

IV. Verifying Initial-state Opacity Using BRG

To the best of our knowledge, no alternative method to the one in Section III-B has been proposed to verify initial-state opacity in bounded PNs. However, such an approach suffers from the well-known *state explosion problem*, since the RG needs to be constructed. To overcome the potential state explosion problem, we propose a new method based on BRG analysis.

A. Basis reachability graph

In the work of Cabasino et al. [12], [15], a compact way to represent the reachability set of a PN is proposed to solve the fault diagnosis problem. Under the assumption that the T_u-induced subnet is acyclic, only part of the reachable markings of the PN, called *basis markings*, are computed, while, all non-basis markings are characterized by a set of linear equations associated with each basis marking. Using the notion of basis markings, the *basis reachability graph* (BRG) is defined. It is an NFA in which each state corresponds to a basis marking and all events are observable. The BRG well preserves the information on the reachability set, as well as on the evolution of the PN, while its structure is usually much more compact than the RG and the state explosion problem may often be avoided. The BRG as proposed in [12], [15] also includes some diagnosis information, which are redundant for opacity verification. Herein we redefine the BRG neglecting such information. Before providing the algorithm for its construction, let us recall some key definitions [12].

Definition 4.1: Given a marking M and an observable transition $t \in T_o$, we define

$$\Sigma(M, t) = \{\sigma \in T_u^* | M[\sigma\rangle M', M' \geq Pre(\cdot, t)\}$$

the set of *explanations* of t at M. ◇

Thus $\Sigma(M, t)$ is the set of unobservable sequences whose firing at M enables t. Among all the explanations, we are interested in finding the minimal ones, i.e., the ones whose firing vector is minimal.

Definition 4.2: Given a marking M and an observable transition $t \in T_o$, we define

$$\Sigma_{min}(M,t) = \{\sigma \in \Sigma(M,t) | \nexists \sigma' \in \Sigma(M,t) : \pi(\sigma') \lneqq \pi(\sigma)\}$$

the set of *minimal explanations* of t at M and $Y_{min}(M,t) = \{y_u \in \mathbb{N}^{n_u} | \exists \sigma \in \Sigma_{min}(M,t) : y_u = \pi(\sigma)\}$ the corresponding set of *minimal e-vectors*. $\diamond$

Algorithm 1 constructs the BRG without diagnoser's states. We denote the BRG as an NFA $\mathcal{B} = (\mathcal{M}_B, E, \Delta)$, where $\mathcal{M}_B$ is the set of basis markings of the LPN, and all events are observable. The transition relation $\Delta \subseteq \mathcal{M}_B \times E \times \mathcal{M}_B$ is determined by the following rule. From a marking M if there is an observable transition t for which an explanation exists, i.e., $\Sigma(M,t) \neq \emptyset$, and the firing of t and one of its minimal explanations lead to M', then an edge from state M to state M' labeled by $\ell(t)$ is added in the BRG, i.e., $(M, \ell(t), M') \in \Delta$.

Algorithm 1 Construction of the BRG

Input: A bounded LPN $G = (N, M_0, E, \ell)$ whose T_u-induced subnet is acyclic.

Output: The BRG $\mathcal{B} = (\mathcal{M}_B, E, \Delta)$

Let $\mathcal{M}_B = \{M_0\}$ and assign no tag to M_0;
while states with no tag exist, **do**
  select a state $M \in \mathcal{M}_B$ with no tag;
  for all t s.t. $\ell(t) \in E$ and $Y_{min}(M,t) \neq \emptyset$, **do**
    for all $y_u \in Y_{min}(M,t)$, **do**
      $M' := M + C_u \cdot y_u + C(\cdot, t)$;
      if $M' \notin \mathcal{M}_B$, **then**
        $\mathcal{M}_B := \mathcal{M}_B \cup \{M'\}$;
        assign no tag to M';
      end if
      $\Delta = \Delta \cup \{(M, \ell(t), M')\}$;
    end for
  end for
  tag node M as "old";
end while
Remove all tags.

Given a word $w \in \mathcal{L}(\mathcal{B}, M_0)$, based on Algorithm 1, if $(M_0, w, M) \in \Delta^*$ then M is the marking reached from M_0 by firing an observable sequence σ_o that produces w and eventually interleaved with some unobservable transitions whose firing is necessary to enable σ_o. Therefore, $\mathcal{M}_B \subseteq R(N, M_0)$.

Notice that to apply BRG, two assumptions are made:

A1) the LPN G is bounded, and

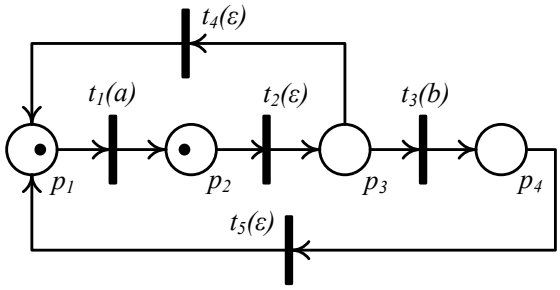


Fig. 1. An LPN whose T_u-induced subnet is acyclic..

A2) the T_u-induced subnet of G is acyclic.

Assumption A1) makes sure that the number of basis markings is finite so that Algorithm 1 can halt, and Assumption A2) is a common technical assumption when partial observation problems, e.g., fault diagnosis or observability, are considered. It allows to use the state equation to characterize the set of markings reached from a basis marking firing unobservable transitions.

Theorem 4.3: [12] Let $G = (N, M_0, E, \ell)$ be an LPN whose T_u-induced subnet is acyclic and $\mathcal{M}_B$ be the set of basis markings. A marking M is reachable if and only if there exists a basis marking $M_b \in \mathcal{M}_B$ such that $M \in \mathcal{U}(M_b)$.

Theorem 4.3 shows that for any reachable marking M, we can always find a basis marking from which M can be reached by firing unobservable transitions. On the other hand, given a basis marking M_b, if M is reachable from M_b by firing unobservable transitions, it is also reachable from M_0. Note that the *if* statement is true, even if Assumption A2) is removed.

As a result of Theorem 4.3, considering the T_u-induced subnet is acyclic a marking is reachable from M_0 if and only if there exists a basis marking M_b such that $M = M_b + C_u \cdot y_u$ allows a non-negative integer solution $y_u \in \mathbb{N}^{n_u}$.

Proposition 4.4: Let $G = (N, M_0, E, \ell)$ be a bounded LPN, and $\mathcal{B} = (\mathcal{M}_B, E, \Delta)$ be its BRG. Given a basis marking $M_b \in \mathcal{M}_B$ and a marking $M \in \mathcal{U}(M_b)$ with $M \neq M_b$, we have $\mathcal{L}(N, \mathcal{U}(M_b)) = \mathcal{L}(\mathcal{B}, M_b)$, and $\mathcal{L}(N, M) \subseteq \mathcal{L}(\mathcal{B}, M_b)$.

Proof: Since $M_b \in \mathcal{U}(M_b)$, $\mathcal{L}(N, \mathcal{U}(M_b)) = \mathcal{L}(N, M_b)$. Therefore, $\mathcal{L}(N, \mathcal{U}(M_b)) = \mathcal{L}(\mathcal{B}, M_b)$. Moreover, as $M \in \mathcal{U}(M_b)$, $\mathcal{L}(N, M) \subseteq \mathcal{L}(N, \mathcal{U}(M_b))$ holds, i.e., $\mathcal{L}(N, M) \subseteq \mathcal{L}(\mathcal{B}, M_b)$. ■

According to Proposition 4.4, the BRG of a PN describes the language generated from reachable markings as well. If a word can be generated from a reachable marking, there must exist a basis marking from which the word can also be generated. In addition, the language generated from a state M_b in the BRG is a superset of the language generated from the marking $M \in \mathcal{U}(M_b)$ with $M \neq M_b$.

Example 4.5: Let us consider the LPN in Fig. 1. Transitions t_1 and t_3 are observable. The labels assigned to them are a and b, respectively. For this net, there are 10 reachable markings and its RG is shown in Fig. 2. However, there

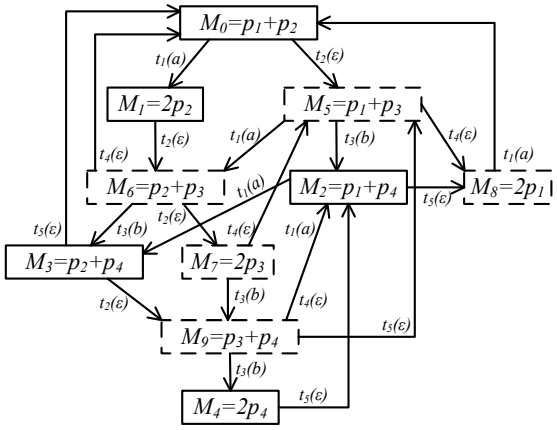


Fig. 2. The RG of the LPN in Fig. 1.

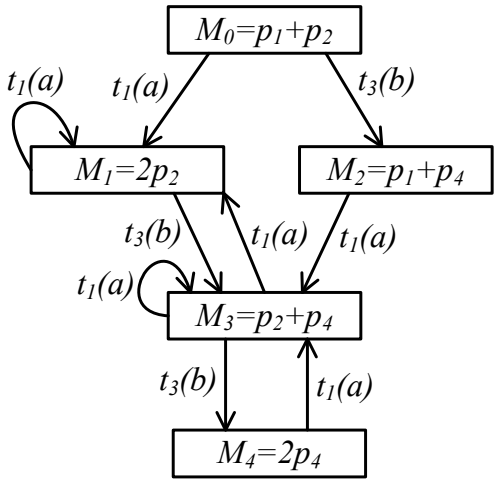


Fig. 3. The BRG of the LPN in Fig. 1.

are only 5 basis markings $\mathcal{M}_B = \{M_0, \ldots, M_4\}$, and the corresponding BRG is shown in Fig. 3. It holds $\mathcal{U}(M_0) = \{M_0, M_5, M_8\}$, $\mathcal{U}(M_1) = \{M_1, M_5, M_6, M_7, M_8\}$, $\mathcal{U}(M_2) = \{M_2, M_8\}$, $\mathcal{U}(M_3) = \{M_0, M_3, M_5, M_8, M_9\}$ and $\mathcal{U}(M_4) = \{M_2, M_4, M_8\}$. Finally, Proposition 4.4 can be easily verified. $\diamond$

B. Reduction to the language containment on the BRG

In this section we show that, when a certain assumption on the secret is satisfied, the language containment problem between the secret and non-secret languages can be reduced to the corresponding problem of the language generated in the BRG. Namely, initial-state opacity of an bounded LPN can be verified by just analyzing the BRG.

Definition 4.6: Let $G = (N, M_0, E, \ell)$ be an LPN, $\mathcal{M}_B$ be the set of basis markings, and S be a secret. The *secret basis marking set* S_B is defined as $S_B = \mathcal{M}_B \cap S$, and the *non-secret basis marking set* $\overline{S}_B$ is defined as $\overline{S}_B = \mathcal{M}_B \cap \overline{S}$. $\diamond$

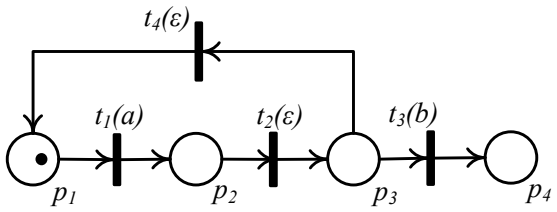


Fig. 4. An LPN that is initial-state opaque wrt $S = \{M_0, M_2\}$.

Given an LPN $G = (N, M_0, E, \ell)$, its BRG $\mathcal{B}$ and the secret S, it always holds $\mathcal{L}(\mathcal{B}, S_B) \subseteq \mathcal{L}(N, S)$ and $\mathcal{L}(\mathcal{B}, \overline{S}_B) \subseteq \mathcal{L}(N, \overline{S})$, since $S_B \subseteq S$ and $\overline{S}_B \subseteq \overline{S}$. Therefore, $\mathcal{L}(\mathcal{B}, S_B) \subseteq \mathcal{L}(\mathcal{B}, \overline{S}_B)$ does not necessarily indicate that $\mathcal{L}(N, S) \subseteq \mathcal{L}(N, \overline{S})$, or vice versa. In other words, by just constructing the BRG, initial-state opacity of the LPN cannot be decided for arbitrary secrets. In the rest of this paper we make the following additional assumption:

A3) $\forall M \in S, \nexists t \in T_u : M[t\rangle M'$ and $M' \notin S$.

In other words, for all secret markings there does not exist an unobservable transition that leads to a non-secret one. This is equivalent to assuming that all markings in the unobservable reach of a secret marking belong to the secret. Note that this assumption can be relaxed by considering all unobservable transitions violating Assumption A3) as observable and then constructing the modified BRG (see [16]). However, the number of states in the modified BRG will increase.

We now prove that if Assumptions A1) to A3) are satisfied, the non-secret language of a net coincides with the non-secret language of its BRG.

Proposition 4.7: Let $G = (N, M_0, E, \ell)$ be an LPN and S be a secret, which satisfy Assumptions A1) to A3). Let $\mathcal{B}$ be the BRG and $\mathcal{M}_B$ be the set of basis markings of G, then we have

$$\mathcal{L}(\mathcal{B}, \overline{S}_B) = \mathcal{L}(N, \overline{S}).$$

Proof: We provide a sketch of the complete proof that can be found in [17]. The $\subseteq$ containment is trivial since $\overline{S}_B \subseteq \overline{S}$. Now we prove $\supseteq$ also holds: if a word is generated from a nonsecret marking, there always exists a nonsecret basis marking from which the word can be generated (otherwise Assumption A3 will be contradicted). ■

Note that for the secret language it does not necessarily hold that $\mathcal{L}(\mathcal{B}, S_B) = \mathcal{L}(N, S)$.

Example 4.8: Let us consider the LPN in Fig. 4. Let $S = \{M_0, M_2\}$ that satisfies Assumption A3). Based on the BRG in Fig. 5(b), we have $S_B = \{M_0\}$ and $\mathcal{L}(N, S_B) = \{\varepsilon, a^n b | n \geq 1\}$. However, $\mathcal{L}(N, S) = \{\varepsilon, a^n b | n \geq 0\}$, i.e., $\mathcal{L}(N, S_B) \subsetneq \mathcal{L}(N, S)$. ◇

However, the following proposition shows that under Assumptions A1) to A3) the language containment between secret and non-secret languages can be verified by just analyzing the BRG.

Proposition 4.9: Let $G = (N, M_0, E, \ell)$ be an LPN and S be a secret, which satisfy Assumptions A1) to A3).

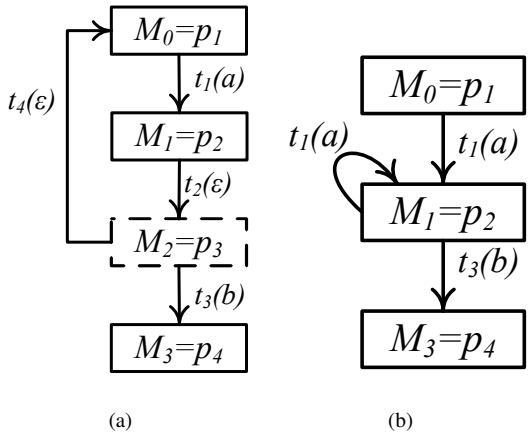


Fig. 5. The RG (a) and the BRG (b) of the LPN in Fig. 4.

Let $\mathcal{B}$ be the BRG and $\mathcal{M}_B$ be the set of basis markings of G. It holds

$$\mathcal{L}(\mathcal{B}, S_B) \subseteq \mathcal{L}(\mathcal{B}, \overline{S}_B) \Leftrightarrow \mathcal{L}(N, S) \subseteq \mathcal{L}(N, \overline{S}).$$

Proof: We provide a sketch of the complete proof that can be found in [17]. According to Propositions 4.4 and 4.7, we just need to prove $\mathcal{L}(N, S_B) \subseteq \mathcal{L}(N, \overline{S}) \Leftrightarrow \mathcal{L}(N, S) \subseteq \mathcal{L}(N, \overline{S})$. The $\Leftarrow$ part is trivial since $\mathcal{L}(N, S_B) \subseteq \mathcal{L}(N, S)$. Now we prove $\Rightarrow$ part also holds. Since for all secret markings M that are not basis markings there exists a secret basis marking M_b such that M is reachable from M_b by firing unobservable transitions, words generated from M can be also generated from M_b, i.e., $\mathcal{L}(N, S \setminus S_B) \subseteq \mathcal{L}(N, S_B) \subseteq \mathcal{L}(N, \overline{S})$. ■

Therefore, under Assumptions A1) to A3), instead of analyzing the RG, we could verify the language containment in the BRG to check if a given LPN is opaque wrt a secret.

Corollary 4.10: Let $G = (N, M_0, E, \ell)$ be an LPN and S be a secret, which satisfy Assumptions A1) to A3). Let $\mathcal{B}$ be the BRG and $\mathcal{M}_B$ be the set of basis markings. G is initial-state opaque wrt S if and only $\mathcal{L}(\mathcal{B}, S_B) \subseteq \mathcal{L}(\mathcal{B}, \overline{S}_B)$.

Proof: It follows from Lemma 3.4 and Proposition 4.9. ■

In other words, Corollary 4.10 proves that the initial-state opacity problem in PNs is equivalent to the language containment problem in the corresponding BRG.

C. Verification of initial-state opacity

In this section we first briefly recall a technique that is used to verify initial-state opacity in automata [8]. Based on the result in the previous section, we show that by applying the technique to the BRG of an LPN, initial-state opacity of the LPN can be effectively verified.

In [8] an automaton called an *initial-state estimator* is proposed based on the reverse automaton. Given an automaton $\mathcal{A}$ without specifying its initial state, the corresponding initial-state estimator $\mathcal{A}_e$ is the observer of its reverse automaton $\mathcal{A}_r$, i.e., the automaton is obtained by revising all arcs in $\mathcal{A}$. In $\mathcal{A}_e$, the state reached by a word w is the set of states from which the word w' can be generated in $\mathcal{A}$, where w' is the reverse of w.

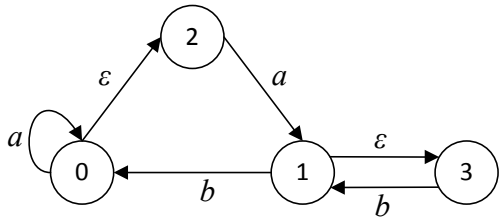
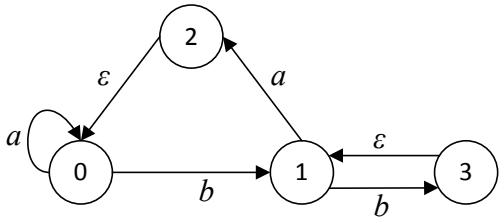
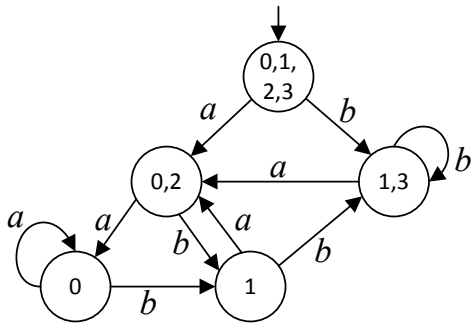


Fig. 6. An automaton $\mathcal{A}$.



(a)



(b)

Fig. 7. The reverse automaton (a) and the initial-state estimator (b) of the automaton in Fig. 6.

Example 4.11: Let us consider the automaton $\mathcal{A}$ in Fig. 6 presented in [10]. Its reverse automaton $\mathcal{A}_r$ and the corresponding observer $\mathcal{A}_e$, i.e., the initial-state estimator, are shown in Figs. 7(a) and 7(b), respectively. Let $w = ab$. In the estimator, the reached state is $\{1\}$, which implies that the set of states that can generate $w' = ba$ in $\mathcal{A}$ is $\{1\}$.

Theorem 4.12: [8] Let $\mathcal{A} = (X, E, \Delta)$ be an automaton and $\mathcal{A}_e = (\mathcal{X}, E, \Delta_e, \mathcal{X}_0)$ be the corresponding initial-state estimator. Given a set of states $Y \subseteq X$, we have $\mathcal{L}(\mathcal{A}, Y) \subseteq \mathcal{L}(\mathcal{A}, \overline{Y})$ if and only if $\forall X_e \in \mathcal{X}$, $X_e \nsubseteq Y$, where $\overline{Y} = X \setminus Y$.

In other words, to verify the language containment the observer of the reverse automaton needs to be constructed. The verification of the language containment $\mathcal{L}(\mathcal{A}, Y) \subseteq \mathcal{L}(\mathcal{A}, \overline{Y})$ has a complexity of $\mathcal{O}(2^{|X|})$. Furthermore, according to Lemma 3.4, initial-state opacity in bounded PNs can be verified by applying Theorem 4.12 to the RG. Therefore, the complexity of verifying initial-state opacity in bounded PNs is $\mathcal{O}(2^{|R(N,M_0)|})$. However, if the PN

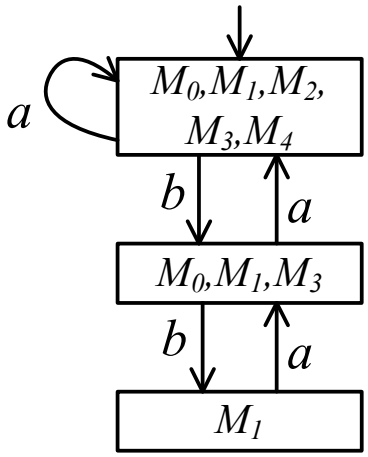


Fig. 8. The initial-state estimator of the BRG in Fig. 3.

and the secret satisfy Assumptions A1) to A3), Theorem 4.12 can be directly applied on BRG.

Corollary 4.13: Let $G = (N, M_0, E, \ell)$ be an LPN, S be a secret that satisfy Assumptions A1) to A3), $\mathcal{B}$ be the BRG of G, $\mathcal{M}_B$ be the set of basis markings, and $\mathcal{B}_e = (\mathcal{X}, E, \Delta_e, \mathcal{X}_0)$ be the corresponding initial-state estimator of $\mathcal{B}$. LPN G is initial-state opaque wrt S if and only if $\forall X_e \in \mathcal{X}$, $X_e \nsubseteq S_B$, where $S_B = \mathcal{M}_B \cap S$.

Proof: Follows from Corollary 4.10 and Theorem 4.12. ■

Therefore, the complexity of using BRG to verify initial-state opacity is $\mathcal{O}(2^{|\mathcal{M}_B|})$. In general, given a bounded PN, it is $|\mathcal{M}_B| \leq |R(N, M_0)|$, therefore, the efficiency of using BRG to verify opacity will not be worse than that of using RG. In particular, when unobservable transitions are considered, the BRG will be smaller than the RG. Moreover, exhaustive enumeration is not needed to compute the BRG. Therefore, BRG brings big advantages over RG for verifying initial-state opacity.

Example 4.14: Consider again the LPN in Fig. 1. The initial-state estimator of its BRG is shown in Fig. 8. Let $S = \{M_0, M_2, M_5, M_8, M_9\}$, then $S_B = \{M_0, M_2\}$ and $\overline{S}_B = \{M_1, M_3, M_4\}$. According to Corollary 4.13, G is initial-state opaque wrt S, since no state of the estimator either coincides with S_B or is strictly contained in it.

V. Conclusions and Future Work

In this paper we propose an efficient approach to verifying initial-state opacity in bounded Petri nets. We proved that under an acceptable assumption on the secret, the verification of initial-state opacity can be transformed into a language containment problem in the basis reachability graph (BRG). Therefore, initial-state opacity can be verified using BRG analysis rather than reachability graph analysis, which provides advantages in terms of computational complexity.

Our future research will be focused on relaxing the assumption on the secret and extend the use of the BRG to the language-based opacity analysis.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China under Grant No. 61374068, 61472295, the Recruitment Program of Global Experts, and the Science and Technology Department Fund, MSAR, under Grant No. 066/2013/A2, and the Italian Ministry of Foreign Affair and International Cooperation (MAECI) under project Robust Decentralised Estimation for large-scale systems (RODEO)-PGR00152.

REFERENCES

[1] N. Busi and R. Gorrieri. A survey on non-interference with Petri nets. In *Lectures on Concurrency and Petri Nets*, pages 328–344. Springer, 2004.

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

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

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

[5] J.W. Bryans, M. Koutny, L. Mazaré, and P.Y. Ryan. Opacity generalised to transition systems. *Int. J. of Information Security*, 7(6):421–435, 2008.

[6] A. Saboori and C.N. Hadjicostis. Notions of security and opacity in discrete event systems. In *46th IEEE Conf. on Decision and Control*, pages 5056–5061, 2007.

[7] E. Badouel, M. Bednarczyk, A. Borzyszkowski, B. Caillaud, and P. Darondeau. Concurrent secrets. *Discrete Event Dynamic Systems*, 17(4):425–446, 2007.

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

[9] R. Jacob, J.J Lesage, and J.M Faure. Opacity of discrete event systems: models, validation and quantification. In *DCDS15*.

[10] A. Saboori and C.N. Hadjicostis. Verification of initial-state opacity in security applications of DES. In *9th Int. Workshop on Discrete Event Systems*, pages 328–333, 2008.

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

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

[13] Y. Tong, Z.W. Li, C. Seatzu, and A. Giua. Verification of current-state opacity using Petri nets. In *2015 American Control Conf.*, 2015.

[14] C.G. Cassandras and S. Lafortune. *Introduction to discrete event systems*. Springer, 2008.

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

[16] M.P. Cabasino, A. Giua, and C. Seatzu. Diagnosability of discrete-event systems using labeled Petri nets. *IEEE Trans. on Automation Science and Engineering*, 11(1):144–153, 2014.

[17] Y. Tong, Z.W. Li, C. Seatzu, and A. Giua. Proofs. `http://www.diee.unica.it/~seatzu/CDC15-proofs.pdf`.