On Two-Way Observer and Its Application to the Verification of Infinite-Step and K-Step Opacity

Xiang Yin and Stéphane Lafortune

***Abstract*— We investigate the verification of the properties of infinite-step opacity and K-step opacity for partially-observed discrete event systems. A system is said to be infinite-step opaque (respectively, K-step opaque) if the intruder can never determine for sure that the system was in a secret state for any instant within infinite steps (respectively, K steps) prior to that particular instant. We derive a new separation principle for state estimates which characterizes the information dependence in this opacity verification problem. A new information structure called the *two-way observer* is proposed. Based on the two-way observer, we provide new algorithms for the verification of infinite-step opacity and the verification of K-step opacity, respectively. We show that the proposed verification algorithms have lower computational complexity than the known algorithms in the literature.**

I. Introduction

Security and privacy are becoming increasingly important issues in the verification and synthesis of networked and cyber-physical systems. In this paper, we consider an important information flow security property called *opacity* in the framework of Discrete Event Systems (DES). Specifically, we consider a system modeled as a finite-state automaton, in which there is a secret the system wants to hide. We say that the system is opaque if the secret cannot be revealed to an intruder that is potentially malicious. The intruder is modeled as an observer that knows the entire structure of the system but can only observe part of the system's behavior.

The notion of opacity was initially introduced in the analysis of cryptographic protocols in [12]. It was extended to the framework of DES in [3], [4]. In the context of DES, various notions of opacity have been studied in order to capture different types of privacy requirements, e.g., language-based opacity [11], current-state opacity [13], initial-state opacity [17], initial-and-final-state opacity [20], K-step opacity [8], [14] and infinite-step opacity [16]. If the original system is not opaque, then one is also interested in enforcing opacity. This problem is referred to as the opacity enforcement problem and it has been studied extensively under various enforcement mechanisms, e.g., using supervisory control theory [1], [7], [15], [18], [24], [25], using dynamic observers [6], [23], [26], using insertion functions [21] and using run-time techniques [8]. Most of the above-mentioned works assume that the system is modeled as a finite-state automaton. Recently, the notion of opacity was extended to other classes of system models, see, e.g., [2], [4], [9], [10], [19].

In this paper, we investigate the *verification* problem of infinite-step opacity and K-step opacity. In contrast to current-state opacity, which requires that the secret not be revealed to the intruder based on the *current* state estimate, infinite-step opacity requires that the secret not be revealed for any instant along the entire observation trajectory up to the present time, based on the observations up to the current time. Similarly, K-step requires that the secret not be revealed within K steps prior to the current instant, based on the observations up to the current time. Although it was shown in [20] that language-based opacity, initial-state opacity, and current-state opacity are equivalent in the sense that they can be mapped to one another in polynomial-time, infinite-step and K-step opacity appear to be incomparable with the above notions. The difference between infinite-step and K-step opacity as compared with current-state opacity (or language-based opacity, initial-state opacity) can be explained intuitively as follows. Current-state opacity only depends on the current state estimate of the system, while infinite-step and K-step opacity allow to do *smoothing*, i.e., to improve state estimation for *earlier* time instants, using observations up to the *present* time. Therefore, infinite-step and K-step opacity are fundamentally more difficult than current-state opacity, language-based opacity, or initial-state opacity.

The notions of infinite-step opacity and K-step opacity were initially studied in [16] and [14], respectively. More specifically, in [14], two different approaches for the verification of K-step opacity were proposed; both of them have the same complexity $O((|E_o|+1)^K \times |E_o| \times 2^{|X|})$, where X and E_o are the set of states and the set of observable events of the system, respectively. For infinite-step opacity, a verification algorithm of complexity $O(|E_o| \times 2^{|X|} \times 2^{|X|^2})$ was provided in [16]. It is worth noting that, for both infinite-step opacity and K-step opacity, it is required that the intruder cannot infer that the system was at a secret state for any *specific instant* in the past. However, in some cases, it is possible that the intruder knows that the system has visited a secret state in the past, although it cannot tell when the secret state was visited. We call a system *trajectory-based* infinite-step (respectively, K-step) opaque if this scenario does *not* occur [14], [16]. Therefore, infinite-step (K-step) opacity is also referred to as *non-trajectory-based* infinite-step (K-step) opacity. Trajectory-based K-step opacity is also referred to as K-step strong opacity in [8], where a verification algorithm is provided. Whether we need to use the trajectory-based

This work was partially supported by NSF grants CCF-1138860 (Expeditions in Computing project ExCAPE: Expeditions in Computer Augmented Program Engineering) and CNS-1421122.

Xiang Yin and Stéphane Lafortune are with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, USA. {xiangyin,stephane}@umich.edu.

notions or the non-trajectory-based notions is application dependent. In this paper, we will focus on the non-trajectory-based notions but the approach proposed is also applicable to the trajectory-based notions.

One of the motivations for studying infinite-step opacity and K-step opacity is that both of these two notions are very useful in some privacy applications. For example, privacy is an important issue in location-based services (LBS); see, e.g., [22]. In the LBS application, the user may what to hide some of her crucial location information (e.g., visiting a bank or a hospital). However, this information may be revealed to an intruder located at the LBS server that keeps tracking the user's queries. Therefore, a formal methodology is needed in order to verify this privacy issue in LBS. It was shown in [22] that checking whether or not the user can always hide her *current* crucial location can be formulated as a current-state opacity verification problem. However, in some cases, the user may also want that the intruder can never be able to infer that she was at a crucial place at some particular instant in the past (e.g., visited bank two days ago). Clearly, current-state opacity is not sufficient to capture this requirement, since the intruder may be able to use future observations to improve its knowledge about the user's location at some particular instant. However, this requirement can be captured using the notions of infinite-step or K-step opacity.

In this paper, we propose new approaches for the verification of infinite-step opacity and K-step opacity. Specifically, the contributions of this paper are as follows. First, we provide a new characterization for the delayed state estimate, which is referred to as the *separation principle*. This result reveals that the information needed in the infinite-step (K-step) opacity verification problem can be decomposed into two mutually independent parts and each of them can be computed individually and effectively. We propose a novel information structure called the Two-Way Observer (TW-observer) in order to capture the independent information described by the separation principle. Then, based on the TW-observer, we provide a new approach for the verification of infinite-step opacity. The new verification algorithm has complexity $O(|E_o| \times 2^{|X|} \times 2^{|X|})$, compared with $O(|E_o| \times 2^{|X|} \times 2^{|X|^2})$ for the previous approach [16]. Finally, we show that the proposed approach can also be used to verify the notion of K-step opacity with complexity $O(\min\{2^{|X|}, |E_o|^K\} \times |E_o| \times 2^{|X|})$. Note that the previous algorithm for checking K-step has a complexity of $O((|E_o|+1)^K \times |E_o| \times 2^{|X|})$ [14]. Therefore, the new algorithm leads to considerable improvement in complexity of verification when K is relatively large.

Due to space constraints, all proofs have been omitted.

II. Opacity Definitions

Let E be a finite set of events and E^* be the set of all finite strings over E including the empty string ϵ. A language $L \subseteq E^*$ is a subset of E^*. We denote by $\overline{L}$ the prefix-closure of L, i.e., $\overline{L} = \{u \in E^* : \exists v \in E^* \text{ s.t. } uv \in L\}$. For any string $s \in E^*$, $|s|$ denotes the length of t. We define $|\epsilon| = 0$.

A DES is modeled as a deterministic finite-state automaton

$$G = (X, E, f, X_0) \tag{1}$$

where X is the finite set of states, E is the finite set of events, $f : X \times E \to X$ is the deterministic transition function where $y = f(x, \sigma)$ means that there exists a transition labeled by event σ from state x to state y, and X_0 is the set of initial states. The transition function f is extended to domain $X \times E^*$ in the usual manner (see, e.g., [5]) and the extended function still denoted by f. The language generated by G from state $x \in X$ is defined by $\mathcal{L}(G, x) = \{s \in E^* : f(x, s)!\}$, where ! means "is defined". For a set of states $Q \subseteq X$, we also define $\mathcal{L}(G, Q) = \cup_{x \in Q} \mathcal{L}(G, x)$. Therefore, the language generated by G is $\mathcal{L}(G) := \mathcal{L}(G, X_0)$. We assume that G is deterministic for the sake of simplicity, but the results developed hereafter can be easily extended to the case where G is nondeterministic.

Given $G = (X, E, f, X_0)$, we denote by $G_R = (X, E, f_R, X)$ the *reversed automaton* of G [20]. Specifically, the transition function $f_R : X \times E \to 2^X$ is defined by: for any state $x, y \in X$ and event $\sigma \in E$, we have $y = f(x, \sigma)$ iff $x \in f_R(y, \sigma)$. Note that G_R is *nondeterministic* in general. Then, for any string $s = \sigma_1 \sigma_2 \ldots \sigma_{|s|} \in E^*$, we denote by s_R the reversed string of s, i.e., $s_R = \sigma_{|s|} \sigma_{|s|-1} \ldots \sigma_1$.

We assume that the intruder, which is modeled as an observer, has the full knowledge of the system's structure, but it can only partially observe the system's behavior. To this end, we assume that the event set E is partitioned into two disjoint subsets, E_o the set of observable events and E_{uo} the set of unobservable events, where $E_o \cup E_{uo} = E$ and $E_o \cap E_{uo} = \emptyset$. The natural projection $P : E^* \to E_o^*$ is defined by

$$P(\epsilon) = \epsilon \text{ and } P(s\sigma) = \begin{cases} P(s)\sigma & \text{if } \sigma \in E_o \\ P(s) & \text{if } \sigma \in E_{uo} \end{cases} \tag{2}$$

The natural projection is also extended to 2^{E^*}, i.e., for any $L \subseteq E^*$, $P(L) = \{t \in E_o^* : \exists s \in L \text{ s.t. } P(s) = t\}$.

Given a set of states $q \in 2^X$ and an observable event $\sigma \in E_o$, we denote by $UR(q)$ is the set of states that can be reached unobservably from some state in q, i.e.,

$$UR(q) := \{x \in X : \exists x' \in q, \exists s \in E_{uo}^* \text{ s.t. } f(x', s) = x\}$$

We also denote by $Next(q, \sigma)$ the set of states that can be reached immediately upon the occurrence of σ, i.e.,

$$Next(q, \sigma) := \{x \in X : \exists x' \in q \text{ s.t. } f(x', \sigma) = x\}$$

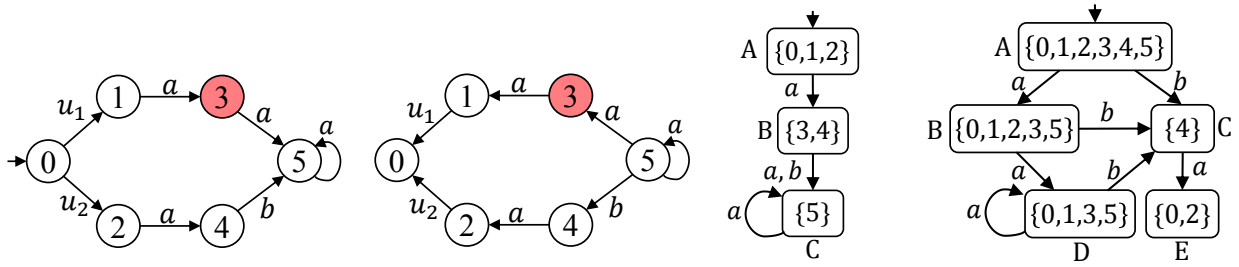
Then the *observer* of G is defined by

$$Obs(G) = (Q_{obs}, E_o, f_{obs}, q_{obs,0}) \tag{3}$$

where $Q_{obs} \subseteq 2^X$, $q_{obs,0} = UR(X_0)$ and for any $q \in 2^X, \sigma \in E_o$, $f_{obs}(q, \sigma) = UR(Next(q, \sigma))$.

We denote by $\hat{X}(s, G)$ the *current-state estimate* associated with observed string $s \in P(\mathcal{L}(G))$ w.r.t. G, i.e.,

$$\hat{X}(s, G) = \{x \in X : \exists x_0 \in X_0, \exists t \in \mathcal{L}(G, x_0) \text{ s.t. } f(x_0, t) = x \wedge P(t) = s\}$$



(a) G (b) G_R (c) $Obs(R)$ (d) $Obs(G_R)$

Fig. 1. System G with $E_o = \{a, b\}$ and $X_S = \{3\}$.

In particular, for any string $s \in P(\mathcal{L}(G))$, we have that $f_{obs}(q_{obs,0}, s) = \hat{X}(s, G)$. Also, we denote by $Obs(G_R) = (Q_{obs,R}, E_o, f_{obs,R}, X)$ the observer of the reversed automaton G_R with initial state X.

Example 1: Consider the automaton G shown in Figure 1(a), where $E_o = \{a, b\}$. The reversed automaton G_R of G is shown in Figure 1(b), where all states are initial states. The observers $Obs(G)$ and $Obs(G_R)$ for automata G and G_R, are shown in Figures 1(c) and 1(d), respectively. For example, for string $u_1 a \in \mathcal{L}(G)$, we have that $P(u_1 a) = a$ and $\hat{X}(a, G) = \{3, 4\} = f_{obs}(q_{obs,0}, a)$.

The system G has a set of secret states, denoted by $X_S \subseteq X$. We assume for simplicity that $X \setminus X_S$ is the set of non-secret states. We say that the system is *K-step opaque* if for any string that leads to a secret state, the intruder, which can observe the occurrences of events in E_o, can never determine for sure that the system is in a secret state at that point using up to K observations thereafter. We recall the formal definition from [14].

Definition 1: (K-Step Opacity). Given system G, set of observable events E_o, set of secret states X_S and non-negative integer $K \in \mathbb{N}$, system G is said to be K-step opaque (w.r.t. E_o and X_S) if

$$\begin{aligned}&(\forall x_0 \in X_0, \forall st \in \mathcal{L}(G, x_0) : f(x_0, s) \in X_S \wedge |P(t)| \leq K)\\ &(\exists x_0' \in X_0, \exists s't' \in \mathcal{L}(G, x_0'))\\ &[f(x_0', s') \notin X_S \wedge P(s') = P(s) \wedge P(t') = P(t)]\end{aligned} \tag{4}$$

When $K \to \infty$, K-step opacity becomes infinite-step opacity. We recall the formal definition from [16].

Definition 2: (Infinite-Step Opacity). Given system G, set of observable events E_o and set of secret states X_S, system G is said to be infinite-step opaque (w.r.t. E_o and X_S) if

$$\begin{aligned}&(\forall x_0 \in X_0, \forall st \in \mathcal{L}(G, x_0) : f(x_0, s) \in X_S)\\ &(\exists x_0' \in X_0, \exists s't' \in \mathcal{L}(G, x_0'))\\ &[f(x_0', s') \notin X_S \wedge P(s') = P(s) \wedge P(t') = P(t)]\end{aligned} \tag{5}$$

Example 2: Consider again the system G in Figure 1(a). Let $X_S = \{3\}$ be the set of secret states. It is easy to verify that Equation (4) does not hold for $K = 1$. By taking $s = u_1 a$ and $t = a$, we know that the only string $s't' \in \mathcal{L}(G)$ such that $P(s') = P(s)$ and $P(t') = P(t)$ is st itself. Intuitively, it says that by observing aa, the intruder will know for sure that the system was in secret state 3 one step earlier. Therefore, G is not 1-step opaque w.r.t. E_o and X_S, which also implies that G is not infinite-step opaque. However, this system is current-state opaque (or 0-step opaque), since the intruder can never determine whether or not the system is currently in a secret state.

In [14] and [16], different approaches for the verification of K-step opacity and infinite-step opacity are provided. Specifically, in [14], two approaches called the state mapping-based approach and the observation sequence-based approach are provided for the verification of K-step opacity; both of them have (worst-case) complexity $O(|E_o| \times (|E_o| + 1)^K \times 2^{|X|})$ [1]. In [16], an algorithm for the verification of infinite-step opacity is proposed by using a bank of initial-state estimators, which has complexity $O(|E_o| \times 2^{|X|} \times 2^{|X|^2})$. The reader is referred to [14], [16] for more details. Hereafter, we will provide a uniform and more efficient approach for the verification of K-step and infinite-step opacity.

III. Delayed State Estimate and Its Characterization

In this section, we first show how infinite-step opacity can be characterized by using the delayed state estimate that was originally proposed in order to characterize K-step opacity [14]. Then we provide a separation principle for the delayed state estimate by dividing it into two independent components.

First, we recall the definition of delayed state estimate from [14].

Definition 3: Let $s = \sigma_1 \sigma_2 \ldots \sigma_n \in P(\mathcal{L}(G))$. Let $K \leq n$ be a non-negative integer. Then the K-delayed state estimate associated with s, denoted by $\hat{X}_{|s|-K}(s)$, is defined as the set of states the system could have been in K steps earlier, after observing s where $|s| \geq K$. Mathematically, we have

$$\begin{aligned}\hat{X}_{|s|-K}(s) :=\{&x \in X : \exists x_0 \in X_0, \exists t_1 t_2 \in \mathcal{L}(G, x_0) \text{ s.t.}\\ &x = f(x_0, t_1) \wedge P(t_1) = \sigma_1 \sigma_2 \cdots \sigma_{n-K}\\ &\wedge P(t_2) = \sigma_{n-K+1} \cdots \sigma_n\}\end{aligned}$$

Clearly, the delayed estimate is a generalization of both the initial-state estimate and the current-state estimate. For any string $s \in P(\mathcal{L}(G))$, $\hat{X}_{|s|-K}(s)$ becomes the initial-state estimate when $K = |s|$ and becomes the current-state

[1] This complexity was originally expressed as $O((|E_o| + 1)^K \times 2^{|X|})$ in [14] because it only considers the number of states in the state estimator structure. In order to obtain the time complexity, the original complexity should be multiplied by $|E_o|$, namely, we also need to consider the number of transitions in the structure.

estimate when $K = 0$. Note that $\hat{X}_{|s|-K}(s)$ is always a non-empty set for any $s \in P(\mathcal{L}(G)), K \leq |s|$.

It was shown in [14] that the system G is K-step opaque, if and only if,

$$\forall s \in P(\mathcal{L}(G)), \forall k \leq \min\{K, |s|\} : \hat{X}_{|s|-k}(s) \not\subseteq X_S \quad (6)$$

Similarly, the next result says that infinite-step opacity can be characterized by the delayed state estimate, if we do not set the delay to a fixed K. For this purpose, we define

$$\hat{X}_{|s|}(st) := \{x \in X : \exists x_0 \in X_0, \exists t_1 t_2 \in \mathcal{L}(G, x_0) \text{ s.t. } x = f(x_0, t_1) \wedge P(t_1) = s \wedge P(t_2) = t\}$$

Proposition 1: The system G is infinite-step opaque (w.r.t. X_S and E_o) if and only if

$$\forall st \in P(\mathcal{L}(G)) : \hat{X}_{|s|}(st) \not\subseteq X_S \quad (7)$$

Observe that for any $st \in P(\mathcal{L}(G))$, the delayed state estimate $\hat{X}_{|s|}(st)$ can never by empty. Computing $\hat{X}_{|s|}(st)$ for a string $st \in P(\mathcal{L}(G))$ is not a easy task, since it not only depends on the information available at the point when s is observed, but also depends on the additional information obtained thereafter from suffix t. Moreover, the length of the suffix t can be unbounded in general. This is also the essential difference between infinite-step opacity and current/initial-state opacity.

Next, we present one of the key results in this paper, which is also referred to as the *separation principle* hereafter. It reveals that for any string $st \in P(\mathcal{L}(G))$, the delayed state estimate $\hat{X}_{|s|}(st)$ consists of two parts that only depend on string s and string t, respectively.

Theorem 1: For any string $st \in P(\mathcal{L}(G))$, we have

$$\hat{X}_{|s|}(st) = \hat{X}(s, G) \cap \hat{X}(t_R, G_R) \quad (8)$$

or equivalently,

$$\hat{X}_{|s|}(st) = f_{obs}(q_{obs,0}, s) \cap f_{obs,R}(X, t_R) \quad (9)$$

We illustrate the above result by the following example.

Example 3: Let us go back to Example 2. Consider strings $s = a$ and $t = a$ such that $st \in P(\mathcal{L}(G))$. We have $t_R = t = a$. Then according to Theorem 1, we know that

$$\begin{aligned} \hat{X}_{|s|}(st) &= f_{obs}(q_{obs,0}, a) \cap f_{obs,R}(X, a) \\ &= \{3, 4\} \cap \{0, 1, 2, 3, 5\} \\ &= \{3\} \subseteq X_S \end{aligned}$$

Therefore, by Proposition 1, we know that G is not infinite-step opaque w.r.t. E_o and X_S.

Theorem 1 has the following important implications. It reveals that given a string s and its suffix t, the delayed state estimate $\hat{X}_{|s|}(st)$ essentially consists of two parts of information: the pre-information obtained by observing s, i.e., $f_{obs}(X_0, s)$ and the post-information obtained thereafter by observing t, i.e., $f_{obs,R}(X, t_R)$. More importantly, these two information sets are mutually independent or *separated*, i.e., $\hat{X}_{|s|}(st)$ can be calculated by simply taking the intersection of the pre-information with the post-information. In other words, computing the post-information does not depend on where the suffix t comes from. It can be simply calculated by using the reversed automaton from initial state X, i.e., we assume that there is no pre-knowledge about where t comes from, since this information will be "taken care of" by $f_{obs}(X_0, s)$. However, $P(\mathcal{L}(G))$ contains an infinite number of strings in general, and for each string in $P(\mathcal{L}(G))$, we need to know at what point we should divide it into its pre-information and its post-information. In other words, we need to build a finite structure in order to capture all strings in $P(\mathcal{L}(G))$ and all possible breakpoints for each string. This idea is formalized by the structure of "two-way observer", which is defined in the next section.

IV. The Two-Way Observer

In this section, we first define the notion of "Two-Way Observer" (TW-Observer), which essentially asynchronously composes the observer of G and the observer of G_R. Then we discuss the properties of the TW-observer.

First, we provide the formal definition of the TW-observer.

Definition 4: The *Two-Way Observer* of G is a deterministic finite-state automaton

$$Obs_{TW}(G) = (Q_{TW}, E_{TW}, f_{TW}, q_{TW,0}) \quad (10)$$

where

- $Q_{TW} \subseteq Q_{obs} \times Q_{obs,R}$ is the set of states;
- $E_{TW} = (E_o \times \{\epsilon\}) \cup (\{\epsilon\} \times E_o)$ is the set of events;
- $q_{TW,0} = (q_{obs,0}, X)$ is the single initial state;
- $f_{TW} : Q_{TW} \times E_{TW} \to Q_{TW}$ is the (deterministic) transition function defined by: for any state $(q_1, q_2) \in Q_{TW}$ and event $\sigma \in E_o$, the following transitions are defined

$$f_{TW}((q_1, q_2), (\sigma, \epsilon)) = (f_{obs}(q_1, \sigma), q_2) \quad (11)$$

$$f_{TW}((q_1, q_2), (\epsilon, \sigma)) = (q_1, f_{obs,R}(q_2, \sigma)) \quad (12)$$

For the sake of simplicity, hereafter, we only consider the reachable part of $Obs_{TW}(G)$.

Intuitively, the TW-observer tracks a string s in $P(\mathcal{L}(G))$ from $q_{obs,0}$ and a reversed string t_R in $Rev(P(\mathcal{L}(G)))$ from X, where $Rev(L) = \{s_R : s \in L\}$. Let (q_1, q_2) be a state reached in $Obs_{TW}(G)$. If $q_1 \cap q_2 \neq \emptyset$, then it means that the above two strings s and t_R "coincide" at some state. In other words, this state could be a "breakpoint" for some string in $\mathcal{L}(G)$, since some strings s' and t' such that $P(s') = s$ and $P(t') = t$ can be "connected" at a state in $q_1 \cap q_2$, i.e., $s't' \in \mathcal{L}(G)$. Before we formalize the above discussion, we introduce some necessary notions. For any string $t \in \mathcal{L}(Obs_{TW}(G))$, we denote by $\tau_1(t) \in E_o^*$ and $\tau_2(t) \in E_o^*$ the first and second components of string t, respectively. For example, if $t = (a, \epsilon)(a, \epsilon)(\epsilon, b)$, then $\tau_1(t) = aa$ and $\tau_2(t) = b$.

Lemma 1: Let $t \in \mathcal{L}(Obs_{TW}(G))$ be a string in the TW-observer and $f_{TW}(q_{TW,0}, t) = (q_1, q_2)$ be the state reached by t. Then we have

$$q_1 \cap q_2 \neq \emptyset \Rightarrow (\exists s \in \mathcal{L}(G))[\tau_1(t)(\tau_2(t))_R = P(s)] \quad (13)$$

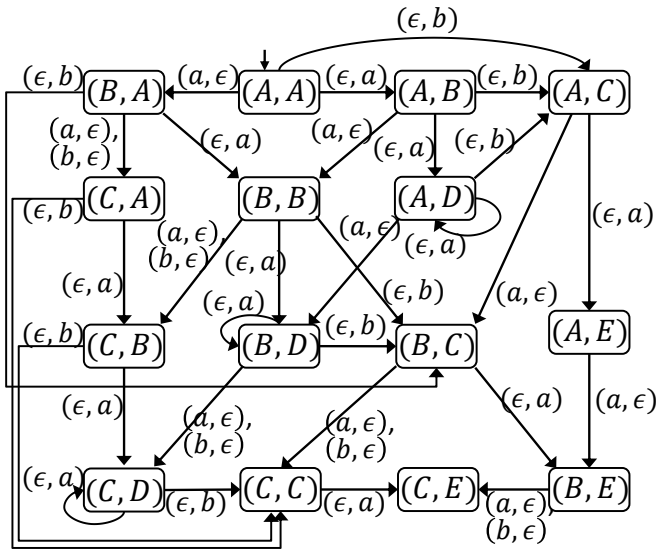


Fig. 2. The two-way observer $Obs_{TW}(G)$ for the system in Figure 1(a)

or equivalently,

$$q_1 \cap q_2 \neq \emptyset \Rightarrow \tau_1(t)(\tau_2(t))_R \in P(\mathcal{L}(G)) \quad (14)$$

Similarly, for any string $s_1 s_2 \in \mathcal{L}(G)$, we can find a corresponding string $t \in \mathcal{L}(Obs_{TW}(G))$ such that the first component of t is $P(s_1)$ and the second component of t is the reversed string of $P(s_2)$. This is formalized by the following lemma.

Lemma 2: For any string $s = s_1 s_2 \in P(\mathcal{L}(G))$, there exists a string $t \in \mathcal{L}(Obs_{TW}(G))$ such that $\tau_1(t) = s_1$ and $(\tau_2(t))_R = s_2$.

The next example illustrates the TW-observer.

Example 4: Again, we consider the system G in Figure 1(a), where $E_o = \{a, b\}$ and $X_S = \{3\}$. The TW-observer $Obs_{TW}(G)$ for this system is shown in Figure 2. For the sake of simplicity, for each state in Q_{TW}, the first and the second components of the state are depicted by using short-hand notation according to Figure 1(c) and Figure 1(d), respectively. For example, state (C, D) represents state $(\{5\}, \{0, 1, 3, 5\})$, which can be reached by string $(a, \epsilon)(b, \epsilon)(\epsilon, a)$. Since $\{5\} \cap \{0, 1, 3, 5\} = \{5\} \neq \emptyset$, by Lemma 1, we know that string $ab(a)_R = aba$ exists in $P(\mathcal{L}(G))$.

V. Verification of Infinite-Step Opacity

In this section, we use the results developed so far and propose a new algorithm for the verification of infinite-step opacity.

According to Theorem 1, we see that the states in the TW-observer essentially capture all possible combinations of $\hat{X}(s, G)$ and $\hat{X}(t_R, G_R)$. Therefore, if the system is infinite-step opaque, then there should not exist a state in $Obs_{TW}(G)$ such that the intersection of its first and second components is a subset of secret states. This idea is formalized by the following theorem, which reveals that, in order to verify infinite-step opacity, it suffices to check whether or not the TW-observer contains a state in which the intersection of the two components is a subset of the set of secret states.

Theorem 2: Let G be the system automaton, E_o be the set of observable events and X_S be the set of secret states. Let $Obs_{TW}(G) = (Q_{TW}, E_{TW}, f_{TW}, q_{TW,0})$ be the TW-Observer of G. Then G is infinite-step opaque w.r.t. E_o and X_S if and only if

$$\forall (q_1, q_2) \in Q_{TW} : q_1 \cap q_2 \not\subseteq X_S \text{ or } q_1 \cap q_2 = \emptyset \quad (15)$$

The following example illustrates how to use the above theorem for the verification of infinite-step opacity.

Example 5: We still consider the system G in Figure 1(a), where $E_o = \{a, b\}$ and $X_S = \{3\}$. The TW-observer $Obs_{TW}(G)$ is shown in Figure 2. We see that state (B, B), which denotes state $(\{3, 4\}, \{0, 1, 2, 3, 5\})$, is reached by string $(a, \epsilon)(\epsilon, a)$ or string $(\epsilon, a)(a, \epsilon)$. Since $\{3, 4\} \cap \{0, 1, 2, 3, 5\} = \{3\} \subseteq X_S$, by Theorem 2, we know that G is not infinite-step opaque.

Remark 1: We discuss the complexity of the above approach for the verification of infinite-step opacity. Clearly, in the worst case, there are at most $2^{|X|} \times 2^{|X|}$ states and $|E_o| \times 2^{|X|} \times 2^{|X|}$ transitions in the TW-observer. Therefore, the worst-case complexity of the proposed algorithm is $O(|E_o| \times 2^{|X|} \times 2^{|X|})$. Notice that the complexity of this TW-observer-based verification algorithm is smaller than the existing algorithm proposed in [16], which is $O(|E_o| \times 2^{|X|} \times 2^{|X|^2})$. It was shown in [16] that the verification of infinite-step opacity is PSPACE-hard. Therefore, it seems highly unlikely that there exists a polynomial-time algorithm for the verification of infinite-step opacity.

VI. Verification of K-Step Opacity

In this section, we discuss the verification of K-step opacity. First, we show how the TW-observer can be used to verify K-step opacity.

Theorem 3: Let G be the system automaton, E_o be the set of observable events and X_S be the set of secret states. Let $Obs_{TW}(G) = (Q_{TW}, E_{TW}, f_{TW}, q_{TW,0})$ be the TW-Observer of G. Then G is K-step opaque w.r.t. E_o and X_S, if and only if, for any string $s \in \mathcal{L}(Obs_{TW}(G))$ such that $f_{TW}(q_{TW,0}, s) = (q_1, q_2)$, we have that

$$[q_1 \cap q_2 \subseteq X_S \wedge q_1 \cap q_2 \neq \emptyset] \Rightarrow |\tau_2(s)| > K \quad (16)$$

Theorem 3 immediately suggests an approach to verify K-step opacity. First, we construct a weighted directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$, where each vertex in $\mathcal{V}$ corresponds to a state in $Obs_{TW}(G)$, each edge in $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ corresponds to a transition in $Obs_{TW}(G)$ and the weight function $w : \mathcal{E} \to \{0, 1\}$ assigns each edge a zero weight if its corresponding event is of the form (σ, ϵ) and a unit weight if its corresponding event is of the form (ϵ, σ). Then we compute the minimum weight of paths from the initial vertex to a vertex which corresponds to state (q_1, q_2) such that $q_1 \cap q_2 \neq \emptyset$ and $q_1 \cap q_2 \subseteq X_S$. If the minimum weight computed is larger than K, then we know that G is K-step opaque. Note that finding the minimum weight can be done in $O(|\mathcal{V}| + |\mathcal{E}|)$ by a breadth-first search. Hence, the worst-time complexity of this approach is $O(|E_o| \times 2^{|X|} \times 2^{|X|})$. Recall that the complexity of the algorithm in [14] is $O(|E_o| \times (|E_o| + 1)^K \times 2^{|X|})$. Also, it worth noting that, in the worst, the delay K can

be as large as $2^{|X|^2}$ [14]. (If K is larger than $2^{|X|^2}$ then it suffices to verify infinite-step opacity.) Therefore, the TW-observer-based approach has considerable improvement for the case where K is relatively large.

In fact, the complexity of the above proposed approach can be further reduced from $O(|E_o| \times 2^{|X|} \times 2^{|X|})$ to $O(\min\{2^{|X|}, |E_o|^K\} \times |E_o| \times 2^{|X|})$ as follows. Since we just to need check whether or not we can reach a "secret-revealing" state (i.e., a state (q_1, q_2) such that $\emptyset \neq q_1 \cap q_2 \subseteq X_S$) from the initial state within K edges with unit value, there is no need to construct the entire TW-observer or the corresponding graph. Instead, we just need to perform a K-weight breadth-first search from the initial state, which yields a "reduced" version of the TW-observer. Namely there is no need to consider states that cannot be reached within K weights from the initial state. By the property of breadth-first search, we know that the number of states of the second component of the reduced TW-observer is bounded by $\min\{|E_o|^K, 2^{|X|}\}$, since state changes in the second component will contribute a unit weight. Therefore, the total complexity of this modified approach is $O(\min\{2^{|X|}, |E_o|^K\} \times |E_o| \times 2^{|X|})$, which is always smaller than the algorithm in [14].

The next example illustrates how to verify K-step opacity.

Example 6: We still consider the system G shown in Figure 1(a), where $E_o = \{a, b\}$ and $X_S = \{3\}$. The weight of transition $(A, A) \xrightarrow{(a,\epsilon)} (B, A)$ is zero and the weight of transition $(B, A) \xrightarrow{(\epsilon,a)} (B, B)$ is one. Moreover, for state (B, B), which denotes state $(\{3, 4\}, \{0, 1, 2, 3, 5\})$, we have that $\{3, 4\} \cap \{0, 1, 2, 3, 5\} = \{3\} \subseteq X_S$. Therefore, the minimum weight computed is 1, which means that G is not 1-step opaque. Moreover, states $(A, D), (B, D), (C, D), (C, E), (B, E)$ and (A, E) will not be considered by using the reduced approach discussed above, since any string that leads to these states contains at least two events of the form of (ϵ, σ).

VII. Conclusion

In this paper, we considered the two information flow properties of infinite-step opacity and K-step opacity. We derived a separated principle and proposed a new structure called two-way observer in order to capture the information flow when analyzing these properties. New algorithms for the verification of these two properties were provided. For infinite-step opacity, we showed that the proposed algorithm is more efficient and has lower complexity than the existing algorithm in the literature. For K-step opacity, we showed that the proposed algorithm is also more efficient and leads to significant improvement when K is relatively large. Moreover, we believe that the separation principle that we established and the notion of TW-observer bring new insights into estimation problems where inferencing about the past is considered. In the future, we plan to extend the notion of TW-observer to synthesize supervisors that enforce infinite-step or K-step opacity.

References

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

[2] B. Bérard, K. Chatterjee, and N. Sznajder. Probabilistic opacity for markov decision processes. *Information Processing Letters*, 115(1):52–59, 2015.

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

[4] J.W. Bryans, M. Koutny, and P. Ryan. Modelling opacity using petri nets. *Electronic Notes in Theo. Comp. Sci.*, 121:101–115, 2005.

[5] C.G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Springer, 2nd edition, 2008.

[6] F. Cassez, J. Dubreil, and H. Marchand. Synthesis of opaque systems with static and dynamic masks. *Formal Methods in System Design*, 40(1):88–115, 2012.

[7] J. Dubreil, P. Darondeau, and H. Marchand. Supervisory control for opacity. *IEEE Trans. Automatic Control*, 55(5):1089–1100, 2010.

[8] Y. Falcone and H. Marchand. Enforcement and validation (at runtime) of various notions of opacity. *Discrete Event Dynamic Syst.: Theo. & Appl.*, pages 1–40, 2014.

[9] C. Keroglou and C.N. Hadjicostis. Initial state opacity in stochastic des. In *18th IEEE Conference on Emerging Technologies & Factory Automation*, pages 1–8. IEEE, 2013.

[10] K. Kobayashi and K. Hiraishi. Verification of opacity and diagnosability for pushdown systems. *J. Applied Mathematics*, 2013, 2013.

[11] F. Lin. Opacity of discrete event systems and its applications. *Automatica*, 47(3):496–503, 2011.

[12] L. Mazaré. Using unification for opacity properties. *Verimag Technical Report*, 2004.

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

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

[15] A. Saboori and C.N. Hadjicostis. Opacity-enforcing supervisory strategies via state estimator constructions. *IEEE Trans. Automatic Control*, 57(5):1155–1165, 2012.

[16] A. Saboori and C.N. Hadjicostis. Verification of infinite-step opacity and complexity considerations. *IEEE Trans. Automatic Control*, 57(5):1265–1269, 2012.

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

[18] S. Takai and Y. Oka. A formula for the supremal controllable and opaque sublanguage arising in supervisory control. *SICE Journal of Control, Measurement, and System Integration*, 1(4):307–311, 2008.

[19] Y. Tong, Z. Li, C. Seatzu, and A. Giua. Verification of current-state opacity using petri nets. In *American Control Conference*, pages 1935–1940, 2015.

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

[21] Y.-C. Wu and S. Lafortune. Synthesis of insertion functions for enforcement of opacity security properties. *Automatica*, 50(5):1336–1348, 2014.

[22] Y.-C. Wu, K.A. Sankararaman, and S. Lafortune. Ensuring privacy in location-based services: An approach based on opacity enforcement. In *WODES*, pages 33–38, 2014.

[23] X. Yin and S. Lafortune. A general approach for solving dynamic sensor activation problems for a class of properties. In *54th IEEE Conference on Decision and Control*, pages 3610–3615, 2015.

[24] X. Yin and S. Lafortune. A new approach for enforcing opacity via supervisory control for partially-observed discrete-event systems. In *American Control Conference*, pages 377–383, 2015.

[25] X. Yin and S. Lafortune. A uniform approach for synthesizing property-enforcing supervisors for partially-observed discrete-event systems. *IEEE Trans. Autom. Contr.*, 2016. Doi 10.1109/TAC.2015.2484359.

[26] B. Zhang, S. Shu, and F. Lin. Maximum information release while ensuring opacity in discrete event systems. *IEEE Transactions on Automation Science and Engineering*, 12(4):1067–1079, 2015.