Verification of Infinite-Step Opacity and Complexity Considerations

Anooshiravan Saboori and Christoforos N. Hadjicostis

Abstract

Motivated by security considerations in applications of discrete event systems, we describe and analyze the complexity of verifying the notion of *infinite-step opacity* in systems that are modeled as non-deterministic finite automata with partial observation on their transitions. Specifically, a system is infinite-step opaque if the entrance of the system state *at any particular instant* to a set of *secret states* remains opaque (uncertain), for the *length* of the system operation, to an intruder who observes system activity through some projection map. In other words, based on observations through this map and complete knowledge of the system model, the intruder can never be certain (and will never be certain) that the system state at any fixed point in time evolves to (or has evolved through) the set of secret states. Infinite-step opacity can be used to characterize the security requirements in many applications, including encryption using pseudo-random generators, coverage properties in sensor networks, and anonymity requirements in protocols for web transactions. We show that infinite-step opacity can be verified via the construction of a set of appropriate *state estimators* and provide illustrative examples. We also establish that the verification of infinite-step opacity is a PSPACE-hard problem.

I. Introduction

Motivated by the increased reliance on shared cyber-infrastructures in many application areas (ranging from defense and banking to health care and power distribution systems), various notions of *security* (in particular, *privacy*) have been receiving attention from researchers. Many such

This material is based upon work supported in part by the U.S. National Science Foundation, under NSF ITR Award 0426831 and NSF CNS Award 0834409. The research leading to these results has also received funding from the European Community Seventh Framework Programme (FP7/2007-2013) under grant agreements INFSO-ICT-223844 and PIRG02-GA-2007-224877. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of NSF or EC.

A. Saboori is with Microsoft; he was with the Coordinated Science Laboratory, and the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign. C. N. Hadjicostis is with the Department of Electrical and Computer Engineering, University of Cyprus, and with the Coordinated Science Laboratory, and the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign. Corresponding author's address: 75 Kallipoleos Avenue, P.O. Box 20537, 1678 Nicosia, Cyprus. E-mail: chadjic@ucy.ac.cy.

notions focus on characterizing the *information flow* from the system to the intruder [1], [2]. *Opacity* falls in this category and aims at determining whether a given system's *secret* behavior (i.e., a subset of the behavior of the system that is considered critical and is usually represented by a predicate) can be revealed to outsiders [3]–[5]. More specifically, this requires that the intruder (modeled as a passive observer of the system's behavior) cannot establish the truth of the predicate, perhaps within some pre-defined time interval.

In our earlier work [5], we considered opacity with respect to predicates that are state-based in discrete event systems (DES) that can be modeled as non-deterministic finite automata with partial observations on their transitions. Assuming that the initial state of the given system is (partially) unknown, we defined the secret behavior of the system to be the *evolution* of the system's state to a subset of secret states S. The intruder has full knowledge of the system model and tracks the observable transitions in the system via the observation of the associated labels. Current-state opacity requires that the secret behavior of the system (i.e., the membership of its current state to the set S) remain opaque (uncertain) until the system enters a state outside the set of secret states S [5]. In [5], we extended this notion of opacity to *K-step opacity* (for $K \geq 0$) by requiring that the entrance of the system state to the set of secret states S at any time during the past K observations remain opaque to the intruder. In other words, in a K-step opaque system, after having observed n observations ($n \geq K$), the intruder cannot determine with certainty that the state of the system $0, 1, \ldots,$ or K observations ago belonged to the set of secret states S. This notion is suitable for cases where there is a bounded delay, after which one does not care if the intruder can infer information about behavior that was previously considered secret (e.g., because the secret transaction has completed or because the intrusion will be detected). One can thus think of K-step opacity as a declassification process in which the secret states are declassified after K observations.

Since in many applications the existence of the above bound K might not be viable, in this work, we extend the notion of K-step opacity to *infinite-step opacity* by allowing the delay K to extend arbitrarily. Specifically, for infinite-step opacity we require that, after having observed an arbitrary sequence of n observations (for any finite n), the intruder cannot determine with

certainty that the state of the system $0, 1, 2, \ldots,$ or n observations ago belonged to the set of secret states S. There are many areas where infinite-step opacity can be used to characterize security requirements and some concrete examples are provided later in this note.

The techniques to verify K-step opacity from [5] cannot be directly used to verify infinite-step opacity because K goes to infinity[1]. Therefore, in order to verify infinite-step opacity, we introduce a novel verification method that uses a combination of current-state estimation [6] and *initial-state estimation* [7]. We analyze the state-complexity of this verification method and show that it is exponential in the square of the number of states of the system. We also show that the verification of infinite-step opacity is a PSPACE-hard problem, which is considered strong evidence that no algorithm can verify this property in polynomial-time [8].

Our work in [5] introduces the notion of K-step opacity for deterministic and non-deterministic automata, respectively, whereas this paper introduces the notion of infinite-step opacity for non-deterministic automata and studies its verification method and its complexity. Apart from our own work in [5], the developments in this paper are also related to other existing security work in the area of DESs. In particular, [3] considers general labeled transition systems and introduces various notions of opacity that, in several cases, are undecidable. References [9] and [10] focus on finite state Petri nets and define opacity (current-state) with respect to state-based predicates. Also, the authors of [4] study labelled trees where each path of the tree encodes a possible execution of the system; by introducing the notion of path equivalence (in terms of observable outputs), [4] uses temporal logic to specify information flow properties such as "agent A does not reveal x (a secret) until agent B reveals y (a password)." This notion, essentially corresponds to 0-step opacity. Our work in [5] and in this paper (i) considers the role of delay (via K-step and infinite-step opacity) in such security requirements which is not present in either [3], [4], [9], or [10], and (ii) studies and solves these problems for the case of finite automata. Due to our assumption regarding the underlying system being a finite automaton, the problem of verifying K-step (or infinite-step) opacity becomes decidable, unlike the case for the general framework

[1]In the sequel, we discuss a (high) complexity method that can be used to verify infinite-step opacity indirectly using the technique in [5].

considered in [3], [9], [10].

The authors of [11] consider multiple intruders modeled as observers with different observation capabilities (namely different natural projection maps) and require that no intruder be able to determine that the actual trajectory of the system belongs to the secret language assigned to that intruder. Assuming that the supervisor can observe/control all events, sufficient conditions for the existence of a supervisor with a finite number of states (i.e., a regular supervisor) are subsequently proposed. The assumptions on the controllability and observability of events are partially relaxed in [12] where the authors consider a single intruder that might observe different events than the ones observed/controlled by the supervisor. In contrast to [11] and [12], opacity in our framework assumes that the states of the system can be partitioned into *secret* and *non-secret* ones; this state-based formulation is what enables us to use various state estimators to verify opacity. Also, note that the notions of opacity introduced here are not considered in [11] and [12], and (as explained later in this note) they cannot be easily captured by the framework of [11], [12].

Related to our work here is also the work in [13] where the authors partition the event set into public level and private level events, and consider the verification of *intransitive non-interference*, a property that captures the allowed information flow (e.g., the occurrence of certain events) from private level events to public level events through a downgrading process. Also, our model of the intruder's capability (in terms of observability power) is different from [13] which makes the two frameworks incomparable. When there is no downgrading process, the notion of non-interference can be translated to an instance of 0-step opacity [5]. Note that in this case, 0-step opacity is a more relaxed notion than non-interference since in a non-interferent system with no down grading process, no information flow from private events to public events is allowed, while opacity allows for some information flow in the system. Also, in general, one cannot formulate the notion of K-step opacity for $K > 0$ in the framework of [13].

II. Preliminaries and Background

Let Σ be an alphabet and denote by Σ^* the set of all finite-length strings of elements of Σ, including the empty string ϵ. A language $L \subseteq \Sigma^*$ is a subset of finite-length strings from Σ^*. For

a string ω, $\overline{\omega}$ denotes the *prefix-closure* of ω and is defined as $\bar{\omega} = \{t \in \Sigma^* | \ \exists s \in \Sigma^* : ts = \omega\}$ where ts denotes the concatenation of strings t and s. The post-string ω/t of ω after $t \in \bar{\omega}$ is defined as $\omega/t = s$, $s \in \Sigma^*$, such that $ts = \omega$. For any string t, $|t|$ denotes the length of t [14].

A DES is modeled in this paper as a non-deterministic finite automaton $G = (X, \Sigma, \delta, X_0)$, where $X = \{0, 1, \ldots, N-1\}$ is the set of states, Σ is the set of events, $\delta : X \times \Sigma \to 2^X$ (where 2^X is the power set of X) is the non-deterministic state transition function, and $X_0 \subseteq X$ is the set of possible initial states. The function δ can be extended from the domain $X \times \Sigma$ to the domain $X \times \Sigma^*$ in the routine recursive manner: $\delta(i, rs) := \bigcup_{j \in \delta(i,r)} \delta(j, s)$ for $r \in \Sigma$ and $s \in \Sigma^*$ with $\delta(i, \epsilon) := i$. The behavior of DES G is captured by $L(G) := \{s \in \Sigma^* \mid \exists i \in X_0, \delta(i, s) \text{ is non-empty}\}$. We use $L(G, i) = \{s \in \Sigma^* | \ \delta(i, s) \text{ is non-empty}\}$ to denote the set of all traces that originate from state i of G (so that $L(G) = \bigcup_{i \in X_0} L(G, i)$).

In general, only a subset Σ_{obs} of the events can be observed. Typically, one assumes that Σ can be partitioned into two sets, the set of observable events Σ_{obs} and the set of unobservable events Σ_{uo} (so that $\Sigma_{obs} \cap \Sigma_{uo} = \emptyset$ and $\Sigma_{obs} \cup \Sigma_{uo} = \Sigma$). The natural projection $P : \Sigma^* \to \Sigma_{obs}^*$ can be used to map any trace executed in the system to the sequence of observations associated with it. This projection is defined recursively as $P(rs) = P(r)P(s),\ r \in \Sigma, s \in \Sigma^*$, with $P(r) = r$ if $r \in \Sigma_{obs}$ and $P(r) = \epsilon$ if $r \in \Sigma_{uo} \cup \{\epsilon\}$ [14].

Upon observing some string $\omega \in \Sigma_{obs}^*$ (sequence of observations), the state of the system might not be identifiable uniquely due to the lack of knowledge of the initial state, the partial observation of events, and/or the non-deterministic behavior of the system. We denote the set of states that the system might reside in *given that ω was observed* as the current-state estimate. The *current-state estimator* (or observer) is a deterministic automaton $G_{0,obs}$ which captures these estimates [6] by having each state of $G_{0,obs}$ be associated with a unique subset of states of the original DES G (so that there are at most $2^{|X|} = 2^N$ states and its initial state is associated[2] with X_0). For more details, refer to [6].

Example 1. Consider the DES G in Figure 1-a with $X_0 = X$. Assuming that $\Sigma_{obs} = \{\alpha, \beta\}$,

[2]The common approach is to associate the current-state estimator's initial state with $UR(X_0)$ instead of X_0, where UR denotes the *unobservable reach* of the set X_0. Our choice of X_0 is not restricted in this way but simplifies the construction and does not change the results.

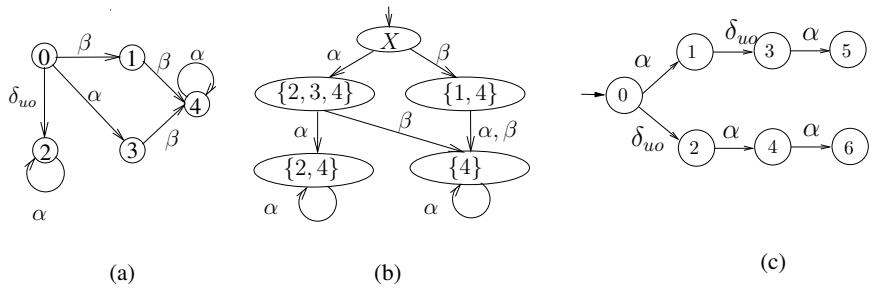

Fig. 1. (a) G; (b) $G_{0,obs}$; (c) DES G in Remark 5.

then the current-state estimator $G_{0,obs}$ in Figure 1-b is constructed as follows. Starting from the initial states in X_0 and observing α, the current state is any of the states in $\{2,3,4\}$; at this new state, the set of possible transitions consists of all possible transitions in G for each of the states in $\{2,3,4\}$. Following this procedure, $G_{0,obs}$ can be completed as in Figure 1-b. ■

Given a finite automaton $G=(X,\Sigma,\delta,X_0)$, a state mapping $m\in 2^{X^2}$ is a set whose elements are pairs of states: the leftmost component of each element (pair) is the *starting state* and the rightmost component is the *ending state*; thus, for a state mapping $m\in 2^{X^2}$, we use $m(1)$ to denote the set of starting states and $m(0)$ to denote the set of ending states. We define the composition operator $\circ : 2^{X^2}\times 2^{X^2}\rightarrow 2^{X^2}$ for state mappings $m_1,m_2\in 2^{X^2}$ as $m_1\circ m_2:=\{(j_1,j_3)|\exists j_2\in X,(j_1,j_2)\in m_1,(j_2,j_3)\in m_2\}$. For any $Z\subseteq X$, we define the operator $\odot : 2^X\rightarrow 2^{X^2}$ as $\odot(Z)=\{(i,i)|i\in Z\}$.

Definition 2 (ω-Induced State Mapping)**.** Given a non-deterministic finite automaton $G=(X,\Sigma,\delta,X_0)$ and a projection map P with respect to the set of observable events Σ_{obs} ($\Sigma_{obs}\subseteq\Sigma$), the ω-induced state mapping after observing string $\omega\in\Sigma_{obs}^*$ is defined as $M(\omega)=\{(i,j)|i,j\in X,\exists t\in\Sigma^*,P(t)=\omega,j\in\delta(i,t)\}$. ■

Note that $M(\omega)=\emptyset$ denotes the fact the sequence of observations ω is not feasible in DES G regardless of its initial state. For $m=\emptyset$, we define $m(1)=m(0)=\emptyset$.

We now briefly review some necessary results and definitions from complexity theory (see [8] for further details). A *problem* is a parameterized question to be answered. An *instance* of a problem is obtained by specifying particular values for all problem parameters. A *decision* problem is one whose answer, depending on the instance, is either "yes" or "no". An algorithm

solves a problem if it produces a correct answer when applied to any instance of the problem.

The class of decision problems that can be solved using space that is polynomial in the size (encoding) of the problem is called PSPACE. A PSPACE-hard problem is a decision problem such that any other decision problem in PSPACE can be reduced to this problem using a polynomial-time algorithm. If a PSPACE-hard problem is in PSPACE, then it is called PSPACE-complete [8]. Showing that a problem is PSPACE-complete is strong evidence that the problem is computationally expensive.

III. Infinite-Step Opacity as the Limiting Case of K-Step Opacity

We first recall the notion of K-step opacity (which was initially defined in [5]).

Definition 3 (K-Step Opacity)**.** Given a non-deterministic finite automaton $G = (X, \Sigma, \delta, X_0)$, a projection map P with respect to the set of observable events Σ_{obs} ($\Sigma_{obs} \subseteq \Sigma$), and a set of secret states $S \subseteq X$, automaton G is K-step opaque (for a nonnegative integer K) with respect to S and P (or (S, P, K)-opaque), if for all $t \in \Sigma^*$, $t' \in \bar{t}$, and $i \in X_0$,

$$\{|P(t)/P(t')| \leq K, \exists j \in S, j \in \delta(i, t'), \delta(j, t/t') \text{ is non-empty}\} \Rightarrow$$
$$\{\exists s \in \Sigma^*, \exists s' \in \bar{s}, P(s) = P(t), P(s') = P(t'),$$
$$\exists i' \in X_0, \exists j' \in \delta(i', s'), j' \in X - S, \delta(j', s/s') \text{ is non-empty}\}.$$ ■

For $t, s \in L(G)$ with $P(s) = P(t)$ we say that t passes through state j when s passes through state j' if there exist $t' \in \bar{t}$, $s' \in \bar{s}$, and $i, i' \in X_0$ such that $P(t') = P(s')$, $j \in \delta(i, t')$, $j' \in \delta(i', s')$ and t/t' and s/s' have continuations from states j and j', respectively. According to Definition 3, DES G is (S, P, K)-opaque if for every string t in $L(G)$ that visits a state j in S within the past K observations (and string t has a continuation from state j), there exists a string s in $L(G)$ with $P(s) = P(t)$ such that when string t passes through the state j in S, string s passes through a state j' in $X - S$ (and string s has a continuation from state j').

Definition 4 (Infinite-Step Opacity)**.** Given a non-deterministic finite automaton $G = (X, \Sigma, \delta, X_0)$, a projection map P with respect to the set of observable events Σ_{obs} ($\Sigma_{obs} \subseteq \Sigma$), and a set of secret states $S \subseteq X$, automaton G is infinite-step opaque with respect to S and P (or (S, P, ∞)-opaque), if for all $t \in \Sigma^*$, $t' \in \bar{t}$, and $i \in X_0$,

$$\{\exists j \in S, j \in \delta(i, t'), \delta(j, t/t') \text{ is non-empty}\} \Rightarrow$$
$$\{\exists s \in \Sigma^*, \exists s' \in \bar{s}, P(s) = P(t), P(s') = P(t'),$$
$$\exists i' \in X_0, \exists j' \in \delta(i', s'), j' \in X - S, \delta(j', s/s') \text{ is non-empty}\}. \quad \blacksquare$$

According to Definition 4, DES G is (S, P, ∞)-opaque if for every string t in $L(G)$ that visits a state j in S (and string t has a continuation from state j), there exists a string s in $L(G)$ with $P(s) = P(t)$ such that when string t passes through the state j in S, string s passes through a state j' in $X - S$ (and string s has a continuation from state j').

Remark 5. The system in Figure 1-c is 2-step opaque with respect to $S = \{1, 6\}$, however, upon observing $\alpha\alpha$, the intruder is certain that, regardless of the state sequence that has occurred, the system has visited a secret state within the last 2 observations (although one cannot pinpoint whether this happened after the first or after the second observation). This system can be considered as insecure if the attacker is only interested in determining whether the system has reached secret states at *any* point during the last K observations (or, more generally, at any point during the observation sequence). A system for which this scenario does not occur will be called *trajectory-based K-step (or infinite-step) opaque*. It is not hard to see that DES G is trajectory-based K-step opaque if and only if for any given sequence of observations ω, there always exists at least one compatible sequence of states such that G visits exclusively non-secret states while generating the last K events in ω.

It can be easily shown that a system that is trajectory-based K-step (infinite-step) opaque is also K-step (infinite-step) opaque; however, as the preceding example demonstrated, the converse is not necessarily true. Note that the essential difference between K-step opacity and trajectory-based K-step opacity is the time at which the state of the system is exposed. Depending on the application, K-step opacity might be a more suitable requirement than trajectory-based K-step opacity for characterizing security requirements. For instance, suppose the DES G in Figure 1-c is a communication protocol for a bank transaction where a user has two options: communicate important account information while at state 1 (secret state) and dummy information while at states 3 and 5 (non-secret states), or communicate dummy information at states 2 and 4 (non-

secret states) and important account information while at state 6 (secret state). If an eavesdropper does not know which of the two options the user has followed (due to the unobservable event δ_{uo}), then (even though she/he knows that important account information has been communicated) she/he does not know when this was done. Therefore, the fact that the system is not 2-step opaque is critical (despite the fact that the system is not trajectory-based 2-step opaque). Related examples with more patient eavesdroppers can be used to motivate infinite-step opacity.

Another example can be found in the context of sensor coverage of mobile agents in various terrains, where one might be interested in characterizing the trajectories of a moving vehicle in a grid of *cells* [15]: if we assume that the movement of the vehicle is monitored by cameras in a subset of cells (i.e., there exists partial coverage) and that certain cells are considered strategic (and hence secret), then the notion of infinite-step opacity can be used to characterize paths that the vehicle can follow without exposing the exact time(s) (measured with respect to the snapshots provided by the cameras) that the vehicle goes through the strategic areas. An extensive analysis of how existing tools can be adjusted to verify tracking properties can be found in [15]. ■

IV. Infinite-Step Opacity Verification

In [5], we introduced a method for verifying K-step opacity (Definition 3) using *K-delay state estimators*. A K-delay state estimator is a deterministic finite automaton which captures the k-delayed state estimates ($0 \leq k \leq K$) associated with a sequence of observations, i.e., estimates of the state of the system k observations ago ($0 \leq k \leq K$) which are consistent with all observations, including the last k observations. This method has state-complexity $O((|\Sigma_{obs}|+1)^K \times 2^N)$ where N is the number of the states of the given automaton in Definition 3. In [5], we also showed that for any $K' > K \geq 2^{N^2} - 1$, K-step opacity and K'-step opacity become equivalent. Since infinite-step opacity is the limiting case of K-step opacity as $K \to \infty$, this implies that K-step opacity for any $K \geq 2^{N^2} - 1$ (e.g., $K = 2^{N^2} - 1$) is equivalent to infinite-step opacity (one direction is obviously true: infinite-step opacity always implies K-step opacity regardless of the value of K; the other direction needs a more careful argument). Therefore, one can verify infinite-step opacity by constructing a $(2^{N^2} - 1)$-delay state estimator and checking whether

$(2^{N^2}-1)$-step opacity holds; however, verifying infinite-step opacity in this way requires an estimator with state-complexity $O((|\Sigma_{obs}|+1)^{2^{N^2}-1}\times 2^N)$ or equivalently, as long as $|\Sigma_{obs}|\geq 1$, $O((|\Sigma_{obs}|+1)^{2^{N^2}})$.

In this section, we introduce a method for verifying infinite-step opacity that has significantly lower complexity than the complexity of the above method. For our development, we first recall the construction of the initial-state estimator in [7] and then discuss how it can be used to verify infinite-step opacity.

A. Initial State Estimation

Given a DES and a sequence of observations ω, the initial-state estimation problem requires the enumeration of all states that belong to the set of initial states X_0 and could have generated this sequence of observations ω. We call this estimate the *initial-state estimate* and denote it by $\hat{X}_0(\omega)$ (note that $\hat{X}_0(\omega)\subseteq X_0$ for all ω). Our earlier work [7] focused on DESs that can be modeled as non-deterministic finite automata under some projection map P with respect to the set of observable events Σ_{obs} ($\Sigma_{obs}\subseteq\Sigma$), and introduced a deterministic finite automaton, called *initial-state estimator* (ISE), to obtain initial-state estimates. The ISE utilizes the notion of a state mapping to capture all the information that can be inferred by any sequence of observations (of finite but arbitrary length) regarding compatible pairs of initial and final states of the system. In particular, each state of the ISE is associated with a unique state mapping and, since the initial state of the system belongs to X_0, the mapping associated with the initial state is the mapping $\odot(X_0)$ (where starting and ending states are identical for all states in X_0). The following composition rule defines subsequent state transitions: given a new observation, the ISE current state transitions into an ISE state whose associated state mapping is the composition of the state mapping associated with the ISE current state and the state mapping induced by the new observation. Continuing in this way we can build a structure which can be shown to provide at any given time (through the state mapping associated with its current state) information about the pairs of system initial states and current states that match the sequence of observations seen so far. Note that this structure is guaranteed to be finite and has at most 2^{N^2} states where N is the number of states of DES G.

Definition 6 (Initial-State Estimator (ISE))**.** Given a non-deterministic finite automaton $G = (X, \Sigma, \delta, X_0)$ and a projection map P with respect to the set of observable events Σ_{obs} ($\Sigma_{obs} \subseteq \Sigma$), the initial-state estimator is the deterministic automaton $G_{\infty,obs} = AC(2^{X\times X}, \Sigma_{obs}, \delta_{\infty,obs}, X_{\infty,0})$ with state set $2^{X\times X}$ (power set of $X \times X$), event set Σ_{obs}, initial state $X_{\infty,0} = \odot(X_0)$, and state transition function $\delta_{\infty,obs} : 2^{X\times X} \times \Sigma_{obs} \to 2^{X\times X}$ defined for $\alpha \in \Sigma_{obs}$ as $m' = \delta_{\infty,obs}(m, \alpha) := m \circ M(\alpha)$, where $m, m' \in 2^{X\times X}$. [AC denotes the states of this automaton that are accessible starting from state $X_{\infty,0}$.] ■

Example 7. Consider the DES G of Figure 1-a with $\Sigma_{obs} = \{\alpha, \beta\}$. Figure 2-a shows the ISE for this system. The initial uncertainty is assumed to be equal to the state space $(X_0 = X)$ and hence the initial state of the ISE is the state mapping $m_0 = \{(0,0), (1,1), (2,2), (3,3), (4,4)\}$. Upon observing α, the next state of the ISE becomes $\{(0,2), (0,3), (2,2), (4,4)\} = m_0 \circ M(\alpha) \equiv m_1$, where $M(\alpha) = \{(0,2), (0,3), (2,2), (4,4)\}$. This means that if the initial state was 0, the current state could be any of the states in $\{2, 3\}$; if the initial state was 2, the current state could only be 2; whereas, if the initial state was 4, the current state would be 4 (on the right of Figure 2-a we use a graphical way to describe the pairs associated with the ISE). If, instead of α, we initially observe β, the ISE transitions to $\{(0,1), (1,4), (3,4)\} = m_0 \circ M(\beta) \equiv m_2$ and implies that β can be observed from states $0, 1$, and 3 and the respective current state can be either 1, 4, and 4. To take into account the observation α followed by observation β, we need to compose the state mapping m_1 with $M(\beta)$ which results in $\{(0,4)\} \equiv m_4$. Using this approach for all possible observations (from each state), the ISE construction can be completed as shown in Figure 2-a. Note that if a sequence of observations ω is not feasible in G, e.g., $\omega = \alpha\beta\beta$, then the state mapping associated with the state reachable in ISE via ω is empty. To avoid cluttering our figures, we do not include this state and the transitions leading to it. ■

B. Verifying Infinite-Step Opacity Using a Bank of ISEs

In order to verify that a system is infinite-step opaque, we need to verify that at any point during the observation process, knowing the sequence of observations before reaching that point, *in addition* to a future observation sequence (that is possible from that point onward), does not

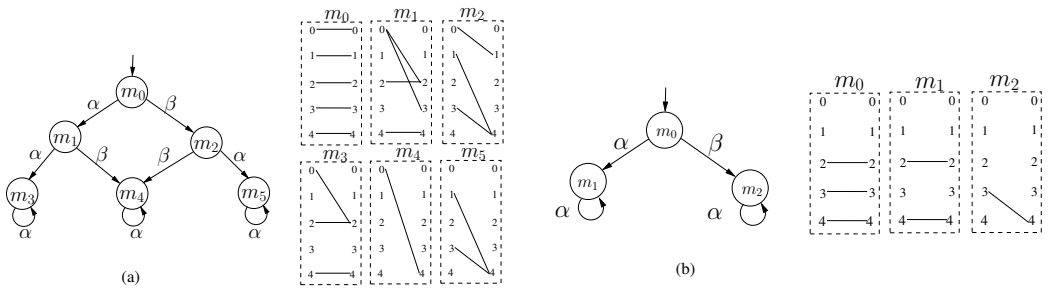

Fig. 2. (a) ISE $G_{\infty,obs}$ for DES G in Figure 1-a; (b) ISE $G^{(4)}_{\infty,obs}$ corresponding to states $\{2,3,4\}$.

(and will not) allow us to determine whether the set of possible states at that point is a subset of the set of secret states. We perform this verification using a two-phase approach: (i) finding all possible estimates of the system's current state along any possible sequence of observations, and (ii) for each point in this trajectory (set of possible system states), calculating the information that can be gained about the state at that point by observation sequences that are possible from that point onward. The first phase can be achieved via a standard current-state estimator [5] (see Example 1). The second phase requires the construction of an ISE-like state estimator for each possible uncertainty about the current-state estimate (which is now used as the initial state estimate for the ISE-like state estimator). In other words, for each set of state estimates $Z \subseteq X$ provided in the first phase, we construct an ISE whose initial state is associated with the state mapping $\odot(Z)$. Clearly, if any of these ISEs contains a state with associated (non-empty) state mapping m such that its set of starting states contains elements only in S (i.e., if $m(1) \subseteq S$), then DES G is not infinite-step opaque. The following theorem formalizes the above discussion and proves that the two-phase approach is correct. The proof can be found in [16].

Theorem 8. *Consider a non-deterministic finite automaton $G = (X, \Sigma, \delta, X_0)$, a projection map P with respect to the set of observable events Σ_{obs} ($\Sigma_{obs} \subseteq \Sigma$), and a set of secret states $S \subseteq X$. For each set of current-state estimates Z_n associated with a state of its current-state estimator $G_{0,obs}$, construct the initial-state estimator $G^{(n)}_{\infty,obs} = AC(2^{X\times X}, \Sigma_{obs}, \delta^{(n)}_{\infty,obs}, X^{(n)}_{\infty,0})$ by setting its initial state $X^{(n)}_{\infty,0}$ to be $\odot(Z_n)$. Then, DES G is (S, P, ∞)-opaque if and only if*

$$\forall n, \forall m \in X^{(n)}_{\infty,obs} : m(1) \nsubseteq S \text{ or } m(1) = \emptyset, \tag{1}$$

where $X^{(n)}_{\infty,obs}$ is the set of states in $G^{(n)}_{\infty,obs}$ that are reachable from $X^{(n)}_{\infty,0} = \odot(Z_n)$ and $m(1)$

denotes the set of starting states of state mapping m. ■

Remark 9. In practice, since the set of initial state estimates can only decrease with additional observations [7], we only need to construct $G^{(n)}_{\infty,obs}$ for Z_n's which have at least one secret state.■

Example 10. In this example, we show that DES G in Figure 1-a is not $(\{3\}, P, \infty)$-opaque. To verify infinite-step opacity we need to first construct the current-state estimator $G_{0,obs}$ as in Figure 1-b. This state estimator has five states $Z_1 = \{4\}$, $Z_2 = \{1, 4\}$, $Z_3 = \{2, 4\}$, $Z_4 = \{2, 3, 4\}$, $Z_5 = \{0, 1, 2, 3, 4\}$; hence, we need to construct five ISEs. Since only state mappings $Z_5 = \{(0, 0), (1, 1), (2, 2), (3, 3), (4, 4)\}$ and $Z_4 = \{(2, 2), (3, 3), (4, 4)\}$ contain the secret state 3, by Remark 9, we only need to construct two ISEs: (i) ISE $G^{(5)}_{\infty,obs}$ with initial state mapping corresponding to Z_5 is indeed the initial-state estimator in Figure 2-a which we constructed previously in Example 7. It can be easily verified that the set of starting states of all (non-empty) state mappings associated with this ISE has states outside the set of secret states. (ii) The ISE $G^{(4)}_{\infty,obs}$ with initial state corresponding to Z_4 is depicted in Figure 2-b (again ignoring the empty state mapping reached via sequences of observations that cannot be generated by G). State $m_2 = \{(3, 4)\}$ in $G^{(4)}_{\infty,obs}$ violates $(\{3\}, P, \infty)$-opacity since its set of starting states only contains state 3 which is a secret state. State m_2 is reachable in $G^{(4)}_{\infty,obs}$ via β from m_0. Moreover, m_0 in this ISE corresponds to the state in $G_{0,obs}$ (in Figure 1-b) that is reached via observation α. Putting these two pieces of information together, we can conclude that observing $\alpha\beta$ reveals that the system has gone through state 3, which is a secret state. ■

The verification of infinite-step opacity using Theorem 8 requires that for each state of the current-state estimator, an ISE-like state estimator be constructed. Since there are at most 2^N states for the current-state estimator, this implies that the complexity of this method is $O(2^N \times 2^{N^2})$. This exponential complexity is not desirable for implementation purposes; as we show in Section IV-C, however, verifying infinite-step opacity is PSPACE-hard and hence it is unlikely that any algorithm can verify this property in polynomial-time [8].

C. Verification of Infinite-Step Opacity is PSPACE-Hard

In order to characterize the complexity class of the infinite-step opacity verification (INF) problem, we first study the complexity of verifying *initial-state opacity* and show that each

instance of the initial-state opacity verification (INI) problem can be reduced (via an algorithm which has complexity polynomial in the number of states of automaton G) to an instance of the INF problem. This proves that the INF problem is at least as hard as the INI problem. Then, we show that the INI problem, for $|\Sigma_{obs}| > 1$, is PSPACE-complete which in turn proves that the INF problem is PSPACE-hard for $|\Sigma_{obs}| > 1$.

Definition 11 (Initial-State Opacity)**.** Given a non-deterministic finite automaton $G = (X, \Sigma, \delta, X_0)$, a projection map P with respect to the set of observable events Σ_{obs} ($\Sigma_{obs} \subseteq \Sigma$), and a set of secret states $S \subseteq X$, automaton G is initial-state opaque with respect to S and P (or (S, P, ∞) initial-state opaque), if for all $i \in X_0 \cap S$ and for all $t \in L(G, i)$ we have

$$\exists j \in X_0 - S, \exists s \in L(G, j), P(s) = P(t). \qquad \blacksquare$$

If none of the secret states of system G is reachable after startup (i.e., if none of the strings in the system can pass through the set of secret states except at startup), then it is not hard to see that infinite-step opacity and initial-state opacity become equivalent. In the following lemma, we use this insight to reduce each instance of the INI problem to an instance of the INF problem. First, we need the following definition.

Definition 12. Given a non-deterministic finite automaton $G = (X, \Sigma, \delta, X_0)$, define the non-deterministic finite automaton $\hat{G} = (\hat{X}, \Sigma, \hat{\delta}, X_0)$ with state set $\hat{X} = X \cup X_0'$ constructed from X by adding duplicates x_0' for each $x_0 \in X_0$ (we denote this by $x_0 \stackrel{d}{\equiv} x_0'$, i.e., $X_0' = \{x_0' | x_0' \stackrel{d}{\equiv} x_0, x_0 \in X_0\}$) and state transition function $\hat{\delta} : \hat{X} \times \Sigma \to 2^{\hat{X}}$ defined for $\alpha \in \Sigma$ and $x \in X$ as $\hat{\delta}(x, \alpha) = \{y | y \in \delta(x, \alpha) - X_0\} \cup \{y' | y \in \delta(x, \alpha) \cap X_0, y \stackrel{d}{\equiv} y'\}$, and for $\alpha \in \Sigma$ and $x' \in \hat{X} - X$ as $\hat{\delta}(x', \alpha) = \hat{\delta}(x, \alpha)$, where $x' \stackrel{d}{\equiv} x$. $\blacksquare$

Lemma 13. *Given a non-deterministic finite automaton $G = (X, \Sigma, \delta, X_0)$, a projection map P with respect to the set of observable events Σ_{obs} ($\Sigma_{obs} \subseteq \Sigma$), and a set of secret states $S \subseteq X$, construct the non-deterministic finite automaton $\hat{G}$ as in Definition 12. Then,*

$$(S, P, \infty) \textit{ initial-state opacity for } G \Leftrightarrow (S \cap X_0, P, \infty)\textit{-opacity for } \hat{G}. \qquad \blacksquare$$

Note that the reduction technique introduced in Lemma 13 clearly has (state or time) complexity that is polynomial in the number of states of G. We prove that the INI problem is PSPACE-

hard using a reduction from the *language containment for non-deterministic finite automata (LC)* problem, which is known to be PSPACE-complete[3] for $|\Sigma| > 1$ [17]. The proof can be found in the Appendix.

Theorem 14. *The INI problem is PSPACE-complete for* $|\Sigma_{obs}| > 1$. ■

Corollary 15. *The INF problem is PSPACE-hard for* $|\Sigma_{obs}| > 1$. ■

Remark 16. While this paper was under review, reference [18] established that the verification of 0-step opacity is PSPACE-complete. ■

V. Conclusion

In this paper, we define, analyze, and characterize the notion of infinite-step opacity as an extension of the notion of K-step opacity [5]. The notion of K-step opacity, for $K \geq 0$, requires that the entrance of a given non-deterministic finite automaton to a set of secret states S, at any time during the past K observations, remain opaque to outsiders. We define infinite-step opacity as the limiting case of K-step opacity as K approaches infinity, and introduce a novel method to verify infinite-step opacity using a current-state estimator and a bank of initial-state estimators. We also establish that the verification of infinite-step opacity is PSPACE-hard.

There are many interesting directions for future research. One important extension is to introduce probabilistic metrics to this framework, e.g., by using information-theoretic metrics to extend the notion of opacity to a probability distribution that captures the likelihood of states given a sequence of observations. Another extension is to employ techniques from supervisory control [14] to design minimally restrictive supervisors for a given discrete event system in a way that enforces infinite-step opacity.

References

[1] R. Focardi and R. Gorrieri, "A taxonomy of trace–based security properties for CCS," in *Proc. of the 7th Workshop on Computer Security Foundations*, June 1994, pp. 126–136.

[2] S. Schneider and A. Sidiropoulos, "CSP and anonymity," in *Proc. of the 4th European Symposium on Research in Computer Security*, September 1996, pp. 198–218.

[3] J. Bryans, M. Koutny, L. Mazare, and P. Ryan, "Opacity generalised to transition systems," *International Journal of Information Security*, vol. 7, no. 6, pp. 421–435, November 2008.

[3]The problem can be solved in polynomial time if $|\Sigma| = 1$.

[4] R. Alur, P. Černý, and S. Chaudhuri, "Model checking on trees with path equivalences," in *Proc. of the 13th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, March 2007, pp. 664–678.

[5] A. Saboori and C. N. Hadjicostis, "Verification of K-step opacity and analysis of its complexity," in *Proc. of the 48th IEEE Conference on Decision and Control*, December 2009, pp. 205–210.

[6] P. E. Caines, R. Greiner, and S. Wang, "Classical and logic–based dynamic observers for finite automata," *IMA Journal of Mathematical Control and Information*, vol. 8, no. 1, pp. 45–80, March 1991.

[7] A. Saboori and C. N. Hadjicostis, "Verification of initial–state opacity in security applications of DES," in *Proc. of the 9th International Workshop on Discrete Event Systems*, May 2008, pp. 328–333.

[8] M. R. Garey and D. S. Johnson, *Computers and Intractability – A Guide to the Theory of NP-Completeness*. Freeman, 1979.

[9] J. W. Bryans, M. Koutny, and P. Y. A. Ryan, "Modelling opacity using Petri nets," *Electronic Notes in Theoretical Computer Science*, vol. 121, pp. 101–115, February 2005.

[10] J. Bryans, M. Koutny, and P. Ryan, "Modelling dynamic opacity using Petri nets with silent actions," in *Formal Aspects in Security and Trust*. Springer, 2005, vol. 173, pp. 159–172.

[11] E. Badouel, M. Bednarczyk, A. Borzyszkowski, B. Caillaud, and P. Darondeau, "Concurrent secrets," *Discrete Event Dynamic Systems*, vol. 17, no. 4, pp. 425–446, December 2007.

[12] J. Dubreil, P. Darondeau, and H. Marchand, "Opacity enforcing control synthesis," in *Proc. of the 9th International Workshop on Discrete Event Systems*, May 2008, pp. 28–35.

[13] N. Hadj-Alouane, S. Lafrance, L. Feng, J. Mullins, and M. Yeddes, "On the verification of intransitive noninterference in multilevel security," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 35, no. 5, pp. 948–958, October 2005.

[14] C. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, 2008.

[15] A. Saboori and C. N. Hadjicostis, "Coverage analysis of mobile agent trajectory via state-based opacity formulations," *Control Engineering Practice (Special Issue on Selected Papers from 2nd International Workshop on Dependable Control of Discrete Systems)*, vol. 19, no. 9, pp. 967–977, September 2011.

[16] A. Saboori, "Verification and enforcement of state-based notions of opacity in discrete event systems," Ph.D. dissertation, University of Illinois at Urbana Champaign, 2010.

[17] S. C. Kleene, "Representation of events in nerve nets and finite automata," in *Automata Studies*, C. E. Shannon and M. McCarthy, Eds. Princeton University Press, 1956, no. 34, pp. 3–41.

[18] F. Cassez, J. Dubreil, and H. Marchand, "Dynamic observers for the synthesis of opaque systems," in *Automated Technology for Verification and Analysis*, October 2009.

APPENDIX

In this section, we provide a formal proof for Theorem 14. First, we need the following definition.

Definition 17 (Language Containment for Non-Deterministic Finite Automata (LC) problem)**.** Given two non-deterministic automata $G_1 = (X_1\ , \Sigma,\ \delta_1, X_{1,0})$ and $G_2 = (X_2, \Sigma, \delta_2, X_{2,0})$ with sets of initial states $X_{1,0}$ and $X_{2,0}$, is $L(G_1) \subseteq L(G_2)$? ■

Proof of Theorem 14: We first prove that the INI problem is in PSPACE for $|\Sigma_{obs}| > 1$. We introduce a polynomial-time algorithm which reduces every instance of the INI problem with $|\Sigma_{obs}| > 1$ to an instance of the LC problem with $|\Sigma| > 1$, and since the LC problem is in PSPACE for $|\Sigma| > 1$, this proves that the INI problem is also in PSPACE for $|\Sigma_{obs}| > 1$. Given a non-deterministic automaton $G = (X, \Sigma, \delta, X_0)$, we construct the non-deterministic automaton $G_o = (X, \Sigma_{obs}, \delta_o, X_0)$ from G as follows. Define the *unobservable reach* for each state x of G as the states reachable from x with a sequence of events which has only one observable event and at least one unobservable event. Then, G_o is constructed from G by removing all unobservable events and connecting each state x to all states in its unobservable reach (each such connection is associated with the observable event designated to that unobservable reach). Computing the unobservable reach takes $O(N^3)$ time [14], where N denotes the number of states of DES G. Next, we construct two non-deterministic automata $G_1 = (X, \Sigma_{obs}, \delta_o, X_{1,0})$ and $G_2 = (X, \Sigma_{obs}, \delta_o, X_{2,0})$ which have the same set of states, event set, and state transition function as G_o, but differ in their set of initial states which are taken to be $X_{1,0} = X_0 \cap S$ and $X_{2,0} = X_0 - S$. Since in constructing these two automata, the structure of G_o is preserved and only the set of initial-states is modified through set intersection, this construction requires $O(N)$ time. It can be shown (but the proof is omitted due to space limitations) that $L(G_1) \subseteq L(G_2)$ if and only if G is (S, P, ∞) initial-state opaque.

Next, we show that the INI problem is PSPACE-hard for $|\Sigma_{obs}| > 1$. We reduce the LC problem with $|\Sigma| > 1$ to an instance of the INI problem with $|\Sigma_{obs}| > 1$ via a polynomial-time algorithm. Given two non-deterministic automata $G_1 = (X_1, \Sigma, \delta_1, X_{1,0})$ and $G_2 = (X_2, \Sigma, \delta_2, X_{2,0})$, define the non-deterministic automaton $G = (X, \Sigma, \delta, X_0)$ with the state set $X = X_1 \cup X_2$, set of initial states $X_0 = X_{1,0} \cup X_{2,0}$, and state transition function $\delta : X \times \Sigma \to 2^X$ given by[4] $\delta(x, \alpha) = \delta_1(x, \alpha)$

[4]Without loss of generality, we assume that $X_1 \cap X_2 = \emptyset$ (one can always rename the states to achieve this).

if $x \in X_1$ and $\delta_2(x, \alpha)$ if $x \in X_2$. Note that the time and state-complexity of constructing G is $O(m^2 + n^2)$ where $m = |X_1|$ and $n = |X_2|$, i.e., polynomial in the number of states of G_1 and G_2. It can be shown (again, we omit the proof due to space limitations) that $L(G_1) \subseteq L(G_2)$ if and only if G is (S, P, ∞) initial-state opaque where $S \equiv X_{1,0}$ and projection map P is with respect to the set of observable events $\Sigma_{obs} = \Sigma$. Since we assume that $\Sigma = \Sigma_{obs}$, this proves that each instance of the LC problem with $|\Sigma| > 1$ can be reduced to an instance of the INI problem with $|\Sigma_{obs}| > 1$ via a polynomial-time algorithm and therefore the INI problem is PSPACE-hard for $|\Sigma_{obs}| > 1$. ■