Opacity-Enforcing Supervisory Strategies via State Estimator Constructions

Anooshiravan Saboori and Christoforos N. Hadjicostis

Abstract—State-based notions of opacity, such as initial-state opacity and infinite-step opacity, emerge as key properties in numerous security applications of discrete event systems. We consider systems that are modeled as partially observed non-deterministic finite automata and tackle the problem of constructing a minimally restrictive opacity-enforcing supervisor (MOES), which limits the system’s behavior within some pre-specified legal behavior while enforcing initial-state opacity or infinite-step opacity requirements. We characterize the solution to MOES, under some mild assumptions, in terms of the supremal element of certain controllable, normal, and opaque languages. We also show that this supremal element always exists and that it can be implemented using state estimators. The result is a supervisor that achieves conformance to the pre-specified legal behavior while enforcing initial-state opacity by disabling, at any given time, a subset of the controllable system events, in a way that minimally restricts the range of allowable system behavior. Although infinite-step opacity cannot be easily translated to language-based opacity, we show that, by using a finite bank of supervisors, the aforementioned approach can be extended to enforce infinite-step opacity in a minimally restrictive way.

Keywords: Supervisory Control, Opacity, Discrete Event Systems, Security

I. INTRODUCTION

Various notions of security and privacy have received considerable attention from researchers. A number of such 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) is kept opaque (i.e., uncertain) to outsiders [3], [4]. More specifically, this requires that the intruder (typically modeled as an observer of partial system behavior) cannot establish the truth of the predicate, perhaps within some pre-specified time interval.

In our earlier work [4]–[6], we considered opacity with respect to state-based predicates in discrete event systems (DESs) that can be modeled as non-deterministic finite automata with partial observation on their transitions. The intruder was assumed to have full knowledge of the system model and be able to track the observable transitions in the system. Assuming that the initial state of the system is (partially) unknown, we defined and analyzed various notions of opacity, including initial-state opacity [5] and infinite-step opacity [6].

In [5], [6], state estimators were used to verify state-based opacity properties. More specifically, an initial-state estimator was used to verify initial-state opacity in [5] and a current-state estimator along with a bank of initial-state estimators were shown to be capable of verifying infinite-step opacity in [6]. In this paper, using the state estimator constructions of [5], [6], we consider the problem of designing a (minimally restrictive) supervisor with the following properties: (i) it limits the system’s behavior within some pre-specified legal behavior, and (ii) it enforces (either initial-state or infinite-step) opacity requirements by disabling, at any given time, the least possible number of events. For enforcing initial-state opacity, we establish that the set of solutions can be characterized as the intersection of controllable, normal, and opaque languages. Using this characterization, we then show that the solution to our problem is the supremal element of such languages. We argue that, under some mild assumptions, the supremal element exists and derive a formulation for it. Moreover, assuming that the given legal behavior is regular (i.e., it can
be described via a finite state machine), we show that the supremal element is also regular. Finally, we propose a procedure that uses an appropriate state estimator to implement this supremal element, effectively integrating the verification and control problems. For enforcing infinite-step opacity, we leverage the results on initial-state opacity and build a finite bank of supervisors that implement minimally restrictive supervisory strategies.

There has been some related work on the design of supervisors to enforce various types of security properties in DES and in the remainder of this section we discuss how this work relates to this paper. The authors of [7] 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 proposed in [7]. The assumptions on the controllability and observability of events are partially relaxed in [8] where the authors consider a single intruder that might observe different events than the ones observed/controlled by the supervisor. Under these assumptions, [8] establishes that a minimally restrictive supervisor always exists, but its regularity depends on the relationship between the set of events observed by the intruder and the sets of events observed/controlled by the supervisor. As we discuss in more detail in later sections of this paper, the supervisor introduced in this paper to enforce initial-state opacity can certainly be formulated in terms of the language framework of [8]. However, our approach leads to a closed form expression for the optimal solution, which is not present in [8], and also allows us to use a state estimator to synthesize the supervisor. The notion of infinite-step opacity (and thus the supervisory control scheme to enforce it) cannot be easily formulated in the framework of [7], [8]; we elaborate this later in the paper.

The authors of [9] model the intruder as a non-passive entity which can override the decision of the supervisor to disable certain events; they are concerned with the problem of intrusion detection and derive sufficient conditions under which a supervisor can detect the presence of an intrusion. Compared to [9], the intruder in our framework is passive (modeled as an observer) and cannot change the system configuration or model.

The authors of [10] consider the problem of designing a minimally restrictive supervisor which enforces non-interference for automata; they partition the set of events into public and private events and define non-interference as the property under which a user who only observes public events cannot determine the occurrence of private events. As we argue in more detail later in the paper, the framework considered in [10] can be translated to an instance of 0-step opacity\(^2\) so that a given system is non-interferent if and only if a transformed system (obtained via an appropriate transformation of the given system) is 0-step opaque. Since the results of this paper can be easily extended to design minimally restrictive supervisors that enforce 0-step opacity (as briefly discussed in Section V), the results in [10] can be generated using the framework of this paper. However, it is not clear if one can formulate the notions of initial-state and infinite-step opacity in the framework of [10].

II. PRELIMINARIES AND BACKGROUND

Let $\Sigma$ be an alphabet and denote by $\Sigma^*$ the set of all finite-length strings of elements of $\Sigma$, including the empty string $\epsilon$. A language $L \subseteq \Sigma^*$ is a subset of finite-length strings from strings in $\Sigma^*$. For a string $\omega$, the prefix-closure of $\omega$ is defined as $\bar{\omega} = \{ t \in \Sigma^* \mid \exists s \in \Sigma^* : ts = \omega \}$ where $ts$ denotes the concatenation of strings $t$ and $s$. The prefix closure $\bar{L}$ of language $L$ is the union of the prefix closures of all strings in $L$. A language is prefix-closed if $L = \bar{L}$. The post-string $\omega/s$ of $\omega$ after $s \in \bar{\omega}$ is defined as $\omega/s = t$ where $t \in \Sigma^*$ and $st = \omega$. The concatenation $L_1L_2$ of two languages $L_1$ and $L_2$ is defined as $L_1L_2 = \{ st \mid s \in L_1, t \in L_2 \}$ [11].

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, $\Sigma$ is the set of events, $\delta : X \times \Sigma \rightarrow 2^X$ (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 $\delta$ can be extended from the domain $X \times \Sigma$ to the

\(^1\)Note that the intruder in our framework can actually be allowed to interfere with uncontrollable events, without having to adjust any of the developments we present.

\(^2\)The notion of 0-step opacity requires that none of the current-state estimates associated with any possible sequence of observations in the system lies entirely within the set of secret states [4].
domain $X \times \Sigma^*$ in the routine recursive manner: \( \delta(i, ts) := \bigcup_{j \in \delta(i, t)} \delta(j, s) \) for \( t \in \Sigma \) and \( s \in \Sigma^* \) with \( \delta(i, \epsilon) := i \). The behavior of DES $G$ is captured by $L(G) := \{ s \in \Sigma^* \mid \exists i \in X_0 \{ \delta(i, s) \neq \emptyset \} \}$. We use $L(G, i)$ to denote the set of all traces that originate from state $i$ of $G$ (so that $L(G) = \bigcup_{i \in X_0} L(G, i)$). The prefix-closed language $E$ is regular if there exists a finite automaton $G$ such that $L(G) = E$ [12], [13].

The product of two non-deterministic automata $G_1 = (X_1, \Sigma_1, \delta_1, X_{01})$ and $G_2 = (X_2, \Sigma_2, \delta_2, X_{02})$ is the automaton $G_1 \times G_2 := AC(X_1 \times X_2, \Sigma_1 \cap \Sigma_2, \delta_{1 \times 2}, X_{01} \times X_{02})$, where $\delta_{1 \times 2}((i_1, i_2), \alpha) := \delta_1(i_1, \alpha) \times \delta_2(i_2, \alpha)$ for $\alpha \in \Sigma_1 \cap \Sigma_2$ and $AC$ denotes the accessible part of the automaton (i.e., the set of states reachable from the (set of) initial state(s) via some string $s \in (\Sigma_1 \cap \Sigma_2)^*$). The construction of the product automaton implies that $L(G_1 \times G_2) = L(G_1) \cap L(G_2)$ [11].

Given a non-deterministic finite automaton $G = (X, \Sigma, \delta, X_0)$, one typically assumes that $\Sigma$ can be partitioned into two sets, the set of observable events $\Sigma_{\text{obs}}$ and the set of unobservable events $\Sigma_{\text{uo}}$ (so that $\Sigma_{\text{obs}} \cap \Sigma_{\text{uo}} = \emptyset$ and $\Sigma_{\text{obs}} \cup \Sigma_{\text{uo}} = \Sigma$). The natural projection $P : \Sigma^* \rightarrow \Sigma_{\text{obs}}$ 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 $P(ts) = P(t)P(s)$, $t \in \Sigma$, $s \in \Sigma^*$, with $P(t) = t$ if $t \in \Sigma_{\text{obs}}$ and $P(t) = \epsilon$ if $t \in \Sigma_{\text{uo}} \cup \{ \epsilon \}$ [12], [13]. $P^{-1}(\omega), \omega \in \Sigma_{\text{obs}}$, denotes the inverse projection, i.e., the set of strings $t \in \Sigma^*$ such that $P(t) = \omega$.

Upon observing some string $\omega \in \Sigma_{\text{obs}}$ (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 $\omega$ was observed as the current-state estimate. The current-state estimator is an automaton $G_{0,\text{obs}}$ which captures these estimates and can be constructed as follows [14]. Each state of $G_{0,\text{obs}}$ is associated with a unique subset of states of the original DES $G$ (so that there are at most $2^{|X|} = 2^N$ states). The initial state of $G_{0,\text{obs}}$ is associated with $X_0$, representing the fact that the initial state could be any state in $X_0$. At any state $Z$ of the estimator ($Z \subseteq X$), the next state upon observing an event $\alpha \in \Sigma_{\text{obs}}$ is the unique state of $G_{0,\text{obs}}$ associated with the set of states that can be reached from (one or more of) the states in $Z$ with a string of events that generates the observation $\alpha$. The following example clarifies this construction. More details can be found in [14].

**Example 1.** Consider the DES $G$ in Figure 1-a with initial state $X_0 = X$. Assuming that $\Sigma_{\text{obs}} = \{ \alpha, \beta \}$, then the current-state estimator $G_{0,\text{obs}}$ in Figure 1-b is constructed as follows. Starting from the initial state $X_0$ and observing $\alpha$, the current state is any of the states in $\{2, 3, 4\}$; at this new state, the set of possible transitions is the union of all possible transitions for each of the states in $\{2, 3, 4\}$. From states in the set $\{2, 3, 4\}$, if $\alpha$ is observed the set of possible states is $\{2, 4\}$, whereas if $\beta$ is observed the set of possible states is $\{4\}$. Following this procedure, $G_{0,\text{obs}}$ can be completed as in Figure 1-b. Note that the state of $G_{0,\text{obs}}$ that is associated with the empty set of states (and that is reached via strings $\omega \in \Sigma_{\text{obs}}$ for which $P^{-1}(\omega) \cap L(G) = \emptyset$) is not drawn in Figure 1-b for clarity purposes. In fact, we can (and will) safely ignore such states since they can never be reached via sequences of observations that are generated by underlying activity in system $G$.

![Fig. 1. (a) G; (b) G_{0,\text{obs}}.](attachment:fig1.png)

A state mapping $m \in 2^{X^2}$ is a set whose elements are pairs of states: the first component of each element (pair) is the starting state and the second 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 $m_1 \circ m_2 := \{(j_1, j_3) \mid j_1, j_2, j_3 \in X \land (j_1, j_2) \in m_1, (j_2, j_3) \in m_2\}$. We define the operator $\oplus : 2^X \rightarrow 2^{X^2}$ as $\oplus(Z) = \{(i, i) \mid i \in Z\}$ where the 2-tuples involve identical elements. We can map any observation $\omega$ of finite but arbitrary length in DES $G$ to a state mapping by using the
mapping $M : \Sigma^* \rightarrow 2^{X^2}$ defined for $\omega \in \Sigma^*$ as $M(\omega) = \{(i, j) | i, j \in X, \exists t \in \Sigma^* \{P(t = \omega, j) \in \delta(i, t)\}\}$. We call $M(\omega)$ the $\omega$-induced state mapping.

In the Ramadge and Wonham framework introduced in [15], it is further assumed that the event set $\Sigma$ can be partitioned into the sets of controllable events ($\Sigma_c$) and uncontrollable events ($\Sigma_{uc}$) so that $\Sigma_c \cap \Sigma_{uc} = \emptyset$ and $\Sigma_c \cup \Sigma_{uc} = \Sigma$. Control is achieved by means of a supervisor which at any given time can disable one or more controllable events. Formally, given a system $G$, a feasible supervisor $\nu_o$ (subscript $o$ denotes the partial observation) for $G$ is a map $\nu_o : P(L(G)) \rightarrow \{\Sigma' \subseteq \Sigma|\Sigma_{uc} \subseteq \Sigma'\}$ which defines the set of events $\Sigma'$ that remain enabled after observing a particular string from the system (note that $\Sigma'$ necessarily includes all uncontrollable events). If we denote the closed-loop system by $\nu_o/G$, the minimally restrictive feasible supervisor problem (MS) is defined as the design of a feasible supervisor $\nu_o$ such that: (i) $L(\nu_o/G) \subseteq E$ for a given (prefix-closed) language $E$ that describes desirable behavior (control objective), and (ii) $L(\nu_o/G)$ is as least restrictive as possible (i.e., for any other feasible supervisor $\nu'_o$ such that $L(\nu'_o/G) \subseteq E$, we have $L(\nu'_o/G) \subseteq L(\nu_o/G)$). If we exclude requirement (ii) (that the supervisor is minimally restrictive) the following theorem from [12], [13] characterizes all solutions to the supervisory control problem for certain prefix-closed languages. In the sequel we assume that $\Sigma_c \subseteq \Sigma_{obs}$.

**Theorem 2 ([12], [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 $\Sigma_{obs}$ ($\Sigma_{obs} \subseteq \Sigma$), a set of controllable events $\Sigma_c$ with $\Sigma_c \subseteq \Sigma_{obs}$, a set of uncontrollable events $\Sigma_{uc} = \Sigma - \Sigma_c$, and a prefix-closed language $K \subseteq L(G)$ with $K \neq \emptyset$, there exists a feasible supervisor $\nu_o$ for $G$ such that $L(\nu_o/G) = K$ if and only if: (i) $K$ is $(L(G), \Sigma_{uc})$–controllable [12], [13] (i.e., $\Sigma_{uc} \cap L(G) \subseteq K$), and (ii) $K$ is $(L(G), P)$–normal [12], [13] (i.e., $K = L(G) \cap P^{-1}(P(K))$).

For any $E \subseteq L(G)$, we define $\mathcal{N}(E)$ ($\mathcal{C}(E)$) to be the set of all prefix-closed sublanguages of $E$ that are normal (controllable). The set $\mathcal{C}E \equiv \mathcal{C}(E) \cap \mathcal{N}(E)$ is closed under union and, hence, there exists a unique supremal element $sup\mathcal{CN}(E)$ under the partial order of set inclusion for this set [12], [13]. We denote $sup\mathcal{CN}(E)$ by $E^{\mathcal{CN}}$. Using this, we can formulate the solution $\nu_o^\mathcal{CN}$ to MS, when limited to normal sublanguages of $E$, as $E^{\mathcal{CN}}$.

In the sequel, the superscript $E^{\mathcal{CN}}$ ($E^{\mathcal{CN}}$) denotes the supremal prefix-closed and $(P(L(G)), \Sigma_{uc})$–controllable ($(L(G), P)$–normal) sublanguage of $E$.

Another approach for defining supervisory control problems is the state-based approach where, instead of specifying the legal behavior as a prefix-closed language $E$, a set of forbidden states is provided via some predicate $R : X \rightarrow \{0, 1\}$ with $R(x) = 0$ capturing the fact that $x \in X$ is a forbidden state. This set of forbidden states needs to be avoided via a state-feedback supervisor $\nu_s : X \rightarrow \{\Sigma' \subseteq \Sigma|\Sigma_{uc} \subseteq \Sigma'\}$ ([12], [13]).

Given the state the system is in, the supervisor determines which controllable events to disable. It can be shown that there exists a state-feedback supervisor such that all states $x$ for which $R(x) = 1$ can be visited under supervision, if and only if $R$ is controllable [12], [13], i.e., if and only if it satisfies the following: (i) if state $m$ satisfies $R$ then $m$ is reachable from the initial state of $G$ via a string of states satisfying $R$, and (ii) at any of the visited states, uncontrollable events take the system to states which again satisfy $R$. If $R$ is not controllable, we can seek a controllable predicate that best approximates $R$ from below. Specifically, we say that predicate $R_1$ refines $R_2$ if for all $x \in X, R_1(x) = 1$ implies $R_2(x) = 1$. Now define $CR(R)$ to be the set of all predicates that are controllable and refine $R$. Then, $CR(R)$ is closed under union and, hence, has a supremal element $supCR(R)$ (denoted by $R^{\mathcal{CR}}$). Let $\nu_s^{\mathcal{CR}}$ be the state-feedback supervisor that synthesizes$^3$ the predicate $supCR(R)$; also denote by $R/G$ the accessible part of automaton $G$ when all the states that do not satisfy $R$ are removed. Then, for any predicate $R$, we have $L(\nu_s^{\mathcal{CR}}/G) = L^{\mathcal{CR}}(R/G)$ [12], [13]. In other words, to find the state-feedback supervisor to synthesize the predicate $supCR(R)$, one can first remove all states that do not satisfy $R$ and then find the supremal controllable sublanguage of the closed-behavior of the remaining state transition diagram.

### III. PROBLEM FORMULATION

#### A. State-Based Notions of Opacity

We recall the definitions of initial-state opacity from [5] and infinite-step opacity from [6].

$^3$Note that if $CR(R) = \emptyset$ then the problem has no solution.
Definition 3 (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 $\Sigma_{\text{obs}} (\Sigma_{\text{obs}} \subseteq \Sigma)$, and a set of secret states $S \subseteq X$, automaton $G$ is initial-state opaque with respect to $S$ and $P$ (or $(S, P, \infty)$ initial-state opaque) if $\forall i \in X_0 \cap S, \forall t \in L(G, i)$,
\[ \exists j \in X_0 - S, \exists s \in L(G, j) \{ P(s) = P(t) \}. \]

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 $\Sigma_{\text{obs}} (\Sigma_{\text{obs}} \subseteq \Sigma)$, and a set of secret states $S \subseteq X$, automaton $G$ is infinite-step opaque with respect to $S$ and $P$ (or $(S, P, \infty)$-opaque), if $\forall t \in X^*, \forall t' \in t, \forall i \in X_0$
\[ \{ \exists j \in S \{ j \in \delta(i, t'), \delta(j, t/t') \neq \emptyset \} \Rightarrow \{ \exists s \in \Sigma^*, \exists s' \in s, \exists j' \in X_0, \exists j' \in \delta(i', s') \} \{ P(s) = P(t), P(s') = P(t'), j' \in X - S, \delta(j', s/s') \neq \emptyset \} \} \].

Remark 5. There are many application areas where initial-state and infinite-step opacity can be used to characterize security requirements of interest (e.g., tracking problems in sensor networks [16]). These applications typically involve systems in which knowing that the system has gone through a secret state but not knowing the exact time at which this occurred does not compromise security or privacy. For instance, suppose the DES $G$ in Figure 2-a 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 while 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 $\delta_0$), then (even though, after observing $\alpha \alpha \alpha$, she/he knows that important account information has been communicated) she/he does not know when this was done.

B. Initial-State Opacity Verification

Definition 6 (Initial-State Estimator (ISE) [5]). 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 $\Sigma_{\text{obs}} (\Sigma_{\text{obs}} \subseteq \Sigma)$, the initial-state estimator is the deterministic automaton

\[ G_{\infty, \text{obs}} = AC((X \times X, \Sigma_{\text{obs}}, \alpha_{\infty, \text{obs}}, X_{\infty, 0}) \text{ with state set } 2^X \times X \text{ (power set of } X \times X) \text{, event set } \Sigma_{\text{obs}}, \text{ initial state } X_{\infty, 0} = \circ(X_0), \text{ and state transition function } \delta_{\infty, \text{obs}} : 2^X \times X \times \Sigma_{\text{obs}} \to 2^X \times X \text{ defined for } \alpha \in \Sigma_{\text{obs}} \text{ as } m' = \delta_{\infty, \text{obs}}(m, \alpha) := m \circ M(\alpha), \text{ where } m, m' \in 2^X \times X. [AC denotes the states of this automaton that are accessible starting from state } X_{\infty, 0}. \]

Clearly, the ISE is a finite structure with at most $2N^2$ states, where $N$ is the number of states of DES $G$. If the ISE contains a reachable state with associated state mapping $m$ such that its set of starting states (is non-empty and) contains elements only in $S$ (i.e., $m(1) \subseteq S$ and $m(1) \neq \emptyset$), then DES $G$ is not initial-state opaque (because the sequence(s) of events that generate a sequence of observations that drives the ISE to state $m$ reveals that the system originated from an initial state in the secret set).

Example 7. Consider the automaton $G$ of Figure 1-a with $\Sigma_{\text{obs}} = \{ \alpha, \beta \}$. Figure 2-b shows the ISE for this system. The initial uncertainty is assumed to be equal to the state space 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 $\alpha$, the next state of the ISE becomes $m_1 = \{(0, 2), (0, 3), (2, 2), (4, 4)\}$ ($m_1 = m_0 \circ M(\alpha)$). Mapping $m_1$ summarizes initial/final state information with its pairs (on the right of Figure 2-b we use a graphical way to describe the pairs associated with the ISE; for example, $\alpha$ can be observed starting from state 0 and ending in state 2 or state 3). Using this approach for all possible observations (from each state), the ISE construction can be completed as shown in Figure 2-b. [This figure does not include the state that corresponds to the empty state mapping and is reached via sequences of observations that cannot be generated by $G$; again, we can (and will) safely ignore this state since it cannot be reached via sequences of
observations that are generated by underlying activity in system $G$. System $G$ is not $(\{1, 3\}, P, \infty)$ initial-state opaque due to the existence of ISE state $m_5 = \{(1, 4), (3, 4)\}$ whose set of starting states $(m_5(1) = \{1, 3\})$ is within $S$. In other words, observing $\beta \alpha(\alpha)^*$ determines the system initial state to have been within the set $\{1, 3\}$. Note that one can use a similar approach to verify that $G$ is $(\{3\}, P, \infty)$ initial-state opaque.

C. Infinite-Step Opacity Verification

One way to verify that a system is infinite-step opaque is 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 (and will not) allow us to determine that the set of possible states at that point is a subset of the set of secret states [6]. Using this intuition, the verification methodology proposed in [6] consists of two phases: (i) find 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), calculate the information that can be gained about the state at that point by possible observation sequences from that point onward. The first phase can be achieved via a standard current-state estimator which captures the estimate of the current state given a sequence of observations. 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).

Example 8. In this example, we show that DES $G$ in Figure 1-a is not $(\{3\}, P, \infty)$-opaque. (Recall that in Example 7, we argued that DES $G$ is $(\{3\}, P, \infty)$ initial-state 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 $Z_0 = \{4\}$, $Z_1 = \{1, 4\}$, $Z_2 = \{2, 4\}$, $Z_3 = \{2, 3, 4\}$, $Z_4 = \{0, 1, 2, 3, 4\}$; hence we need to construct five ISEs with initial states $\{(4, 4)\}$, $\{(1, 1), (4, 4)\}$, $\{(2, 2), (4, 4)\}$, $\{(2, 2), (3, 3), (4, 4)\}$, and $\{(0, 0), (1, 1), (2, 2), (3, 3), (4, 4)\}$, respectively. However, among these initial states only the ones corresponding to $Z_3$ and $Z_4$ contain the secret state 3. This implies that we only need to construct two ISEs: (i) The ISE $G^{(4)}_{\infty, obs}$ with initial state mapping corresponding to $Z_4$ is indeed the initial-state estimator in Figure 2-b which we considered in Example 7. It can be easily verified that the set of starting states of all state mappings associated with this ISE has states outside the set of secret states. (ii) The ISE $G^{(3)}_{\infty, obs}$ with initial state corresponding to $Z_3$ is depicted in Figure 3 (again, in the figure we do not include the state that corresponds to the empty state mapping and is reached via sequences of observations that cannot be generated by $G$). State $m_2 = \{(3, 4)\}$ in $G^{(3)}_{\infty, obs}$ 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 $G^{(3)}_{\infty, obs}$ via $\beta$ 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 $\alpha$. 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.

D. Minimally Restrictive Opacity-Enforcing Feasible Supervisor Problem (MOES)

Opacity is defined to be a property of the states of the given finite automaton; however, the application of supervisory control to the system modifies the original structure of the automaton and hence its states. The remedy to this problem is to find a way to map states of the supervised system to states of the original system and, hence, re-define the set of secret states for the system under supervision to include all those states that are mapped to secret states in the original system.

Definition 9 (Opacity for Supervised System). Given a non-deterministic finite automaton $G = (X, \Sigma, \delta, X_0)$, a projection map $P$ with respect to the set of observable events $\Sigma_{obs}$ ($\Sigma_{obs} \subseteq \Sigma$), and a set of secret states $S \subseteq X$, we say that the non-deterministic automaton $G' = (X', \Sigma', \delta', X'_0)$ is $(S, P, \infty)$ initial-state opaque ($(S, P, \infty)$-opaque) with respect to $G$ if $G_p = G' \times G \equiv (X_p, \Sigma_p, \delta_p, X_{0p})$ is $(S_p, P, \infty)$ initial-
state opaque \((\langle S_p, P, \infty \rangle\text{-opaque})\), where \(X_{0p} = \{(x'_o, x_o) \mid x'_o \in X'_o, x_o \in X_o\}\) and \(S_p = \{(x', x) \in X_p \mid x \in S\}\).

Using Definition 9 (replacing \(G'\) with \(\nu_0/G\) in Definition 9), initial-state (infinite-step) opacity is enforced under supervision if \(\nu_0/G\) (the supervised system) is \((S, P, \infty)\) initial-state opaque \((\langle S, P, \infty \rangle\text{-opaque})\) with respect to \(G\). Note that this definition requires \(\nu_0/G\) to be regular. A feasible supervisor that achieves this property is called an opacity-enforcing feasible supervisor for the system and is denoted by \(\nu_{op}\).

**Definition 10** (MOES). Given a non-deterministic finite automaton \(G = (X, \Sigma, \delta, X_0)\), a projection map \(P\) with respect to the set of observable events \(\Sigma_{obs}\), a set of secret states \(S \subseteq X\), a prefix-closed and regular language \(E \subseteq L(G)\), and a set of controllable events \(\Sigma_c, \Sigma_c \subseteq \Sigma_{obs}\), find an opacity-enforcing feasible supervisor \(\nu_{op}\) for \(G\) such that (i) \(L(\nu_{op}/G) \subseteq E\), and (ii) \(L(\nu_{op}/G)\) is as large as possible (with respect to set inclusion).

The MOES problem that requires enforcing initial-state opacity is denoted with \(\text{MOES}^0\) and the MOES problem that requires enforcing infinite-step opacity is denoted with \(\text{MOES}^\infty\).

### IV. Solution to \(\text{MOES}^0\)

**A. Characterizing the Solutions to \(\text{MOES}^0\)**

The solutions to \(\text{MOES}^0\) can be characterized using machinery that already exists in the literature on supervisory control (e.g., [12], [15]). For our analysis in this section, \(G = (X, \Sigma, \delta, X_0)\) is a non-deterministic finite automaton associated with the natural projection map \(P\) (with respect to the set of observable events \(\Sigma_{obs}\), \(\Sigma_{obs} \subseteq \Sigma\)) and a set of controllable events \(\Sigma_c (\Sigma_c \subseteq \Sigma)\).

**Definition 11** (Language-Based Definition of Initial-State Opacity). Language \(K \subseteq L(G)\) is \((S, P, \infty)\) initial-state opaque with respect to \(G\) if for all \(i \in X_0 \cap S, \forall t \in L(G, i) \cap K,\)

\[ \exists j \in X_0 - S, \exists s \in L(G, j) \cap K \{ P(s) = P(t) \}. \]

The following lemma relates Definition 11 to Definition 9 for a regular language \(K\). The proof can be found in [17].

**Lemma 12.** Given a prefix-closed and regular language \(K \subseteq L(G)\) and the non-deterministic finite automaton \(G_K = (X_K, \Sigma_K, \delta_K, X_{0K})\) such that \(L(G_K) = K\), \(K\) is \((S, P, \infty)\) initial-state opaque with respect to \(G\) if and only if \(G_K\) is \((S, P, \infty)\) initial-state opaque with respect to \(G\).

Using the notion of initial-state opacity for languages, we now characterize all opacity-enforcing feasible supervisors and derive the solution to \(\text{MOES}^0\).

**Theorem 13.** Given a prefix-closed and regular language \(K \subseteq L(G), K \neq \emptyset\), and assuming \(\Sigma_c \subseteq \Sigma_{obs}\), there exists an opacity-enforcing feasible supervisor \(\nu_{op}^0\) for \(G\) such that \(L(\nu_{op}^0/G) = K\), if and only if: (i) \(K\) is \((L(G), \Sigma_{uc})\text{-controllable}\); (ii) \(K\) is \((L(G), P)\text{-normal}\); (iii) \(K\) is \((S, P, \infty)\) initial-state opaque with respect to \(G\).

**Proof:** Follows from Theorem 2 and Lemma 12.

For any \(E \subseteq L(G)\), define \(P^0(E)\) to be the set of prefix-closed sublanguages of \(E\) that are initial-state opaque. Then, using Theorem 13, for any supervisor \(\nu_{op}^0\) that enforces initial-state opacity we have \(L(\nu_{op}^0/G) \subseteq \mathcal{CN}P^0(E) := \mathcal{C}(E) \cap \mathcal{N}(E) \cap P^0(E)\).

**MOES** requires the minimally restrictive opacity-enforcing feasible supervisor, and since \(\text{MOES}^0\) assumes that \(\Sigma_c \subseteq \Sigma_{obs}\), Theorem 13 can be used to characterize the solution to \(\text{MOES}^0\) as the supervisor \(\nu_{op}^{1\mathcal{CN}P^0}\) such that \(L(\nu_{op}^{1\mathcal{CN}P^0}/G) = \sup \mathcal{CN}P^0(E) \equiv E^{1\mathcal{CN}P^0}\). In the next section, we prove that such supremal element exists and provide a formulation for it.

**B. Properties of Initial-State Opaque Languages**

In this section, through various lemmas and theorems (whose proofs can be found in the [17]), we characterize the language \(E^{1\mathcal{CN}P^0}\) and, hence, obtain the solution to \(\text{MOES}^0\).

**Lemma 14.** For any language \(E \subseteq L(G)\), we have \(P^0(E) = \{K \subseteq E \mid K = K, K \subseteq L(G) \cap P^{-1}(P(K \cap L(G, X_0 - S)))\}\).

**Lemma 15.** \(P^0(E)\) is non-empty and closed under arbitrary unions; in particular, the supremal element \(\sup P^0(E)\) exists in \(P^0(E)\).

Assuming that \(E\) is prefix-closed, the following theorem derives a formulation for the supremal element \(E^{1P^0}\) of \(P^0(E)\).

**Theorem 16.** For any prefix-closed language \(E \subseteq L(G), \) we have \(E^{1P^0} = E \cap P^{-1}(P(E \cap L(G, X_0 - S)))\).

**Corollary 17.** Normality is preserved under \(1P^0\) operator for a prefix-closed and normal language \(E \subseteq L(G)\).
We complete the analysis on the properties of initial-state opaque languages by considering the controllability condition.

**Lemma 18.** Initial-state opacity is preserved under the $\uparrow^{CN}$ operator for any prefix-closed and normal language $E \subseteq L(G)$. ■

The following theorem derives a formulation for $E^{\uparrow^{CN}P_0}$.

**Theorem 19.** Given a prefix-closed language $E \subseteq L(G)$, we have
\[
E^{\uparrow^{CN}P_0} = L(G) \cap P^{-1}((P((E^{\uparrow N})^{\uparrow P_0}))^{\uparrow C_a}).
\]

**Remark 20.** Note that in Definition 11, $K$ is not assumed to be prefix-closed; however, Lemma 12, Theorem 13, and Theorem 19 hold only when $K$ is prefix-closed. ■

**C. Implementing the Solution to MOES$^0$ using the Initial-State Estimator**

MOES$^0$ requires the minimally restrictive opacity-enforcing feasible supervisor $\nu_{op}$ that can enforce the legal behavior described via the prefix-closed language $E$. Theorem 13 states that the solution to MOES$^0$ boils down to $E^{\uparrow^{CN}P_0}$ and Theorem 19 characterizes this solution as
\[
E^{\uparrow^{CN}P_0} = L(G) \cap P^{-1}(P(E^{\uparrow P_0}))^{\uparrow C_a}
\]
(assuming that $E$ is normal). Observe that Theorem 16 characterizes $E^{\uparrow P_0} = E \cap P^{-1}(P(E \cap L(G, X_0 - S)))$ which implies that $E^{\uparrow P_0}$ can be implemented using projection and intersection operations on languages. Also [18] provides a formulation for the supremal controllable sublanguage $E^{\uparrow C}$ (assuming that $E$ is prefix-closed); using concatenation, intersection and complementation operations on languages. For regular languages, all these operations can be implemented using operations on finite automata [11]. Since in MOES$^0$ $E$ is assumed to be regular, $E^{\uparrow^{CN}P_0}$ can be obtained (and, hence, the solution to MOES$^0$ can be implemented) via operations on the automata describing $E$ and $G$, which implies that $E^{\uparrow^{CN}P_0}$ is regular. We now point out that the ISE construction can be used for enforcing initial-state opacity and propose an algorithm that efficiently integrates the verification and control problems. In the sequel we assume, without loss of generality, that $E$ is normal (if this assumption is not satisfied, we can always first compute $E^{\uparrow N}$ using the results in [18]).

**Algorithm A:** Consider a non-deterministic finite automaton $G = (X, \Sigma, \delta, X_0)$, with a set of observable events $\Sigma_{obs}$ ($\Sigma_{obs} \subseteq \Sigma$), a set of controllable states $\Sigma_c$ ($\Sigma_c \subseteq \Sigma_{obs}$), a set of secret states $S \subseteq X$, and a prefix-closed, normal and regular language $E \subseteq L(G)$, $E \neq \emptyset$, describing the legal behavior. We can obtain a minimally restrictive supervisor $\nu_{s}^{\uparrow CR}$ for $G$ that enforces initial-state opacity with respect to the natural projection map $P : \Sigma \rightarrow \Sigma_{obs}$ via the following steps: (i) Construct automaton $G_E = (X_E, \Sigma, \delta_E, X_{0E})$ such that $L(G_E) = E$. (ii) Construct $G_p = G_E \times G = (X_p, \Sigma, \delta_p, X_{0p})$; define $S_p \equiv \{(x, e, x) \in X_p | x \in S\}$. (iii) Construct the ISE $G_{\infty,obs}^{(p)} = (X_{\infty,obs}^{(p)}, \Sigma_{obs}, \delta_{\infty,obs}^{(p)}, X_{\infty,0}^{(p)})$ corresponding to $G_p$. (iv) Construct the state-feedback supervisor $\nu_{s}^{\uparrow CR}$ for $G_{\infty,obs}^{(p)}$ that avoids all states in $G_{\infty,obs}^{(p)}$ for which the set of starting states of the associated state mapping (is non-empty and) contains no state outside $S_p$; for this, first define the predicate $R : X_{\infty,obs}^{(p)} \rightarrow \{0, 1\}$ as $R(m) = 0$, if $m(1) \subseteq S_p$ and $m \neq \emptyset$, and $R(m) = 1$, otherwise. Then, construct the accessible part $R/G_{\infty,obs}^{(p)}$ of the ISE $G_{\infty,obs}^{(p)}$ when all the states that do not satisfy $R$ are removed, and find the supremal controllable sublanguage of the closed-behavior of the remaining state transition diagram. ■

**Theorem 21.** Given a prefix-closed, normal and regular language $E \subseteq L(G)$, $E \neq \emptyset$, the control action of the solution $\nu_{op}^{\uparrow^{CN}P_0}$ to MOES$^0$ after observing $\omega$ is the same as the control action of the state-feedback supervisor $\nu_{s}^{\uparrow CR}$ (as synthesized by Algorithm A) at the state reached in $G_{\infty,obs}^{(p)}$ via $\omega$. ■

**Proof:** Proof is provided in [17]. ■

**Example 22.** Consider the DES $G$ in Figure 1-a. Assume that $\Sigma_c = \{\beta\}$. Also suppose that $E = L(G)$ meaning that the only requirement for the supervisor is to enforce initial-state opacity. This implies that steps (i)-(iii) of Algorithm A can be replaced by the
construction of the ISE for $G$. Figure 2-b depicts this ISE for $G$ and denotes it by $G_{\infty, \text{obs}}$. As mentioned in Example 7, this system is not $(\{1, 3\}, P, \infty)$ initial-state opaque due to the existence of state $m_5$ in the ISE. To obtain the minimally restrictive feasible supervisor that enforces $(\{1, 3\}, P, \infty)$ initial-state opacity, following step (iv) of Algorithm A, we first remove from $G_{\infty, \text{obs}}$ the states that violate initial-state opacity, in this case, $m_5$; Figure 4-a depicts the accessible part of the remaining automaton $R/G_{\infty, \text{obs}}$. Next, following step (v) of Algorithm A, we check for the controllability condition. At state $m_2$, $\alpha$ is disabled which is an uncontrollable event. Hence, access to state $m_2$ should be rejected earlier, which is accomplished by disabling $\beta$ at state $m_0$. Figure 4-b depicts the automaton associated with the supremal controllable sublanguage of the automaton in Figure 4-a. Based on this, the supervisor does not allow observing $\beta$ as the first observation. Indeed, observing $\beta \alpha$ determines the system initial state to be within the set $\{1, 3\}$, which consists exclusively of secret states (and, hence, violates initial-state opacity).

Remark 23. The notion of initial-state opacity can be translated to a version of opacity as studied in [7], [8] by defining the secret language to be $K = \bigcup_{i \in X_0 \cap S} L(G, i)$. Using this approach, DES $G$ is $(S, P, \infty)$ initial-state opaque if and only if it is opaque with respect to $K$, i.e., iff $P(K) \subseteq P(L(G) - K)$. Therefore, the supervisory control problem in this paper can be translated to an instance of the supervisory control problem in [7], [8] which is solved via an iterative algorithm (though this algorithm is shown to terminate in one step). In this paper, we have provided a closed-form solution to this problem and also implemented the solution via the construction of the ISE. As mentioned earlier, this integrates the verification and control problems since one will typically have to use the ISE for verifying initial-state opacity. Also, the setting we consider here allows us to limit the system behavior within the legal language $E$ (which is not present in [7], [8]).

The authors of [8] also consider the more general case when the intruder observes the set of events $\Sigma_i \subseteq \Sigma$ which is not necessarily the same as the set of events controlled (observed) by the supervisor $\Sigma_c$ ($\Sigma_o$). The extension of our solution to the cases when $\Sigma_i \subseteq \Sigma_c \subseteq \Sigma_o$ and $\Sigma_c \subseteq \Sigma_o \subseteq \Sigma_i$ is straightforward and follows similar steps as [8]. The case when $\Sigma_c \subseteq \Sigma_i \subseteq \Sigma_o$ is more involved and is the topic of ongoing research. However, following a similar approach as in [8], it might be possible to show that the solution can be formulated as a state-based feedback supervisor for the product of the ISE (constructed assuming that the set of observable events is $\Sigma_i$) and the plant.

V. SOLUTION TO MOES$^\infty$

In order to verify infinite-step opacity, we introduced in Section III-C a method which consists of two phases:

(i) The first phase constructs a standard current-state estimator and, as part of verifying infinite-step opacity, we need to ensure that none of these current-state estimates lies entirely within the set of secret states. In our earlier work [4], we called this property 0-step opacity. If 0-step opacity is violated, we can follow an approach similar to the one in Section IV, to obtain a current-state estimator (instead of an initial-state estimator) and then construct a state-feedback supervisor that enforces 0-step opacity (instead of initial-state opacity). Without loss of generality, we assume that the system under consideration is 0-step opaque (otherwise, one can always design an optimal supervisor that enforces 0-step opacity and then consider the controlled system under this supervisor as the new system to be controlled).

(ii) The second phase constructs, for each current-state estimate $Z \subseteq X$ provided in the first phase, an ISE whose initial state is associated with the state mapping $\circ(Z)$. We argued that if any of these ISEs contains a state with associated (non-empty) state mapping $m$ such that the set of starting states $m(1)$ contains elements only in $S$ (i.e., $m(1) \subseteq S$ and $m(1) \neq \emptyset$), then DES $G$ is not infinite-step opaque. Therefore, one can enforce infinite-step opacity by prohibiting sequences of observations which reach such ISE states in one of the associated ISEs. This requirement can be equivalently described as a MOES$^0$ problem where the intruder knows that the initial state of the system is $Z$. We denote this special case of the MOES$^0$ problem as MOES$^0_Z$ where subscript $Z \subseteq X$ represents that the initial uncertainty about the initial state is $Z$ (as opposed to MOES$^0$ where this uncertainty is taken to be $X_0$). Given any discrete event system $G$ with $N$ states, assume that the current-state estimator has
\( M := |X_{0,\text{obs}}| \leq 2^M \) states, which we denote by \( X_{\text{obs}} = \{Z_0, \ldots, Z_{M-1}\}, Z_q \subseteq X, 0 \leq q \leq M - 1 \). We therefore obtain servers that solve MOES\(_Z\), \( 0 \leq q \leq M - 1 \), as \( \nu^{\text{CNP}}_{op,q} \). Recall that for any solution \( \nu^{\text{CNP}}_{op} \) to MOES\(_0\), we have an equivalent state-feedback supervisor \( \nu^{\text{CR}}_{s,q} \). Thus, for any solution \( \nu^{\text{CNP}}_{op,q} \) to MOES\(_Z\), the equivalent state-feedback supervisor is denoted by \( \nu^{\text{CR}}_{s,q} \). The following algorithm formalizes the construction of \( \nu^{\text{CR}}_{s,q}, 0 \leq q \leq M - 1 \).

**Algorithm B** Consider a non-deterministic finite automaton \( G = (X, \Sigma, \delta, X_0) \), with a set of observable events \( \Sigma_{\text{obs}} (\Sigma_{\text{obs}} \subseteq \Sigma) \), a set of controllable events \( \Sigma_c (\Sigma_c \subseteq \Sigma_{\text{obs}}) \), a set of secret states \( S \subseteq X \), and a prefix-closed, normal and regular language \( E \subseteq L(G), E \neq \emptyset \), describing the legal behavior. Assume (without loss of generality) that \( G \) is 0-step opaque. We can obtain a bank of state-feedback supervisors \( \nu^{\text{CR}}_{s,q} \) for \( G \) that enforce initial-state opacity with respect to the natural projection map \( P : \Sigma \to \Sigma_{\text{obs}} \) via the following steps: (i) Construct automaton \( G_E = (X_E, \Sigma, \delta_E, X_0E) \) such that \( L(G_E) = E \). (ii) Construct \( G_p = G_E \times G = (X_p, \Sigma, \delta_p, X_0p) \) and define \( S_p = \{ (x, x) \in X_p | x \in S \} \). (iii) Construct the current-state estimator \( G_{0,\text{obs}}^{(p)} = (X_{\text{obs}}^{(p)}, \Sigma_{\text{obs}}, \delta_{\text{obs}}^{(p)}, X_{0,\text{obs}}^{(p)}) \) corresponding to \( G_p \). (iv) For each set of current-state estimates \( Z_q \) associated with a state of the current-state estimator \( G^{(p)}_{0,\text{obs}} \), construct the initial-state estimator \( G_{\infty,\text{obs}}^{(p)} = (X_{\infty,\text{obs}}^{(p)}, \Sigma_{\text{obs}}, \delta_{\infty,\text{obs}}^{(p)}, X_{\infty,\text{obs}}^{(p)}) \) by setting its initial state \( X_{\infty,0}^{(p)} \) to be \( \circ(Z_q) \). (v) For each ISE \( G_{\infty,\text{obs}}^{(p)}, 0 \leq q \leq M - 1 \), where \( M \) denotes the number of states of the current-state estimator \( G^{(p)}_{0,\text{obs}} \), construct the state-feedback supervisor \( \nu^{\text{CR}}_{s,q} \) for \( G_{\infty,\text{obs}}^{(p)} \) that avoids all states in \( G_{\infty,\text{obs}}^{(p)} \) for which the set of starting states of the associated state mapping contains no state outside \( S_p \), for this, first define the predicate \( R : X_{\infty,\text{obs}}^{(p)} \to \{0, 1\} \) as \( R(m) = 0 \), if \( m(1) \subseteq S_p \) and \( m \neq \emptyset \), and \( R(m) = 1 \), otherwise. Then construct the accessible part \( R/G_{\infty,\text{obs}} \) of the ISE \( G_{\infty,\text{obs}}^{(p)} \) when all the states that do not satisfy \( R \) are removed, and find the supremal controllable sublanguage of the closed behavior of the remaining state transition diagram.

Following the approach in Theorem 21, it can be shown that the control action of the solution \( \nu^{\text{CNP}}_{op,q} \) to MOES\(_Z\), after observing \( \omega \) and assuming that the initial uncertainty is \( Z_q \) is the same as the control action of the state-feedback supervisor \( \nu^{\text{CR}}_{s,q} \) (as synthesized by Algorithm B) at the state reached in \( G_{\infty,\text{obs}}^{(p)} \) via \( \omega \). In the sequel, we define the state of the supervisor \( \nu^{\text{CR}}_{s,q} \) as the current state of the automaton \( \nu^{\text{CR}}_{s,q} / G_{\infty,\text{obs}}^{(p)} \).

We use the bank of state-feedback supervisors \( \nu^{\text{CR}}_{s,q}, 0 \leq q \leq M - 1 \), associated with each state of the current-state estimator, to enforce infinite-step opacity as follows. Following a sequence of observations \( \omega \), we enable the state-feedback supervisor \( \nu^{\text{CR}}_{s,q} \) associated with the state \( Z_q \) of the current-state estimator (\( Z_q \) is reachable in the current-state estimator via \( \omega \)). Once the state-feedback supervisor \( \nu^{\text{CR}}_{s,q} \) is enabled, it will stay enabled for the remainder of the operation of the system. Note that a state-feedback supervisor \( \nu^{\text{CR}}_{s,q} \) starts operating from its (unique) initial state and its current state is updated according to new observations. This guarantees that the membership of the system’s state to the set of secret states (at the point where the sequence of observations \( \omega \) is made) will be kept opaque for all possible subsequent observations. Since the control action of the state-feedback supervisor \( \nu^{\text{CR}}_{s,q} \) is merely determined by its state, we simply need to track its current state based on the observed events. Upon observing a new label \( \alpha \), we need to first update the current state of all enabled state-feedback supervisors, and also enable a new supervisor to ensure that the membership of the system’s state to the set of secret states (at the point where the sequence of observations \( \omega \alpha \) is made) will be kept opaque for all possible future observations. As a result, after observing the sequence of observations \( \omega, |\omega| \) state-feedback supervisors are enabled and the overall control action is defined to be the intersection of the individual control actions of these \( |\omega| \) supervisors.

**Theorem 24.** Consider a non-deterministic finite automaton \( G = (X, \Sigma, \delta, X_0) \), with a set of observable events \( \Sigma_{\text{obs}} (\Sigma_{\text{obs}} \subseteq \Sigma) \), a set of controllable events \( \Sigma_c (\Sigma_c \subseteq \Sigma_{\text{obs}}) \), a set of secret states \( S \subseteq X \), and a prefix-closed, normal and regular language \( E \subseteq L(G), E \neq \emptyset \), describing the legal behavior. Assume that the system is 0-step opaque and construct the current-state estimator \( G_{0,\text{obs}}^{(p)} = (X_{\text{obs}}^{(p)}, \Sigma_{\text{obs}}, \delta_{\text{obs}}^{(p)}, X_{0,p}) \) and the bank of state-feedback supervisors \( \nu^{\text{CR}}_{s,q}, 0 \leq q \leq M - 1 \), associated with the bank of initial-state estimators \( G_{\infty,\text{obs}}^{(p)} = \)
(X^{(p,q)}_{\infty, \text{obs}}, \Sigma_{\text{obs}}, G^{(p,q)}_{\infty, \text{obs}}, X^{(p,q)}_{\infty, 0})$, $0 \leq q \leq M - 1$, where $M$ denotes the number of states of the current-state estimator $G^{(p)}_{0, \text{obs}}$ (as in Algorithm B). Also assume that upon observing the sequence of observations $\omega = \alpha_0 \alpha_1 \ldots \alpha_n$, the sequence of states visited by the current-state estimator $G^{(p)}_{0, \text{obs}}$ is given by $Z_{q_0} = X_{0p}$, $Z_{q_1}$, ..., $Z_{q_n}$, $Z_{q_n+1}$, $0 \leq q_i \leq M - 1$, $0 \leq i \leq n + 1$, which we denote by $Z_{q_0} \overset{\alpha_0}{\rightarrow} Z_{q_1} \overset{\alpha_1}{\rightarrow} \ldots Z_{q_n} \overset{\alpha_n}{\rightarrow} Z_{q_{n+1}}$. Then:

(i) The control action of the solution $\nu^{\text{CR}}_{\text{op}}$ to MOES$^\infty$ after observing the sequence of observations $\omega$ is

$$
\nu^{\text{CR}}_{\text{op}}(\omega) = \nu^{\text{CR}}_{s, q_0}(z_0) \cap \nu^{\text{CR}}_{s, q_1}(z_1) \cap \ldots \cap \nu^{\text{CR}}_{s, q_n}(z_n),
$$

where $z_i$, $0 \leq i \leq n$, is the state in $G^{(p,q)}_{\infty, \text{obs}}$ that is reached from its initial state $X^{(p,q)}_{\infty, 0}$ via $\alpha_i \alpha_{i+1} \ldots \alpha_n$, i.e., $z_i = G^{(p,q)}_{\infty, \text{obs}}(X^{(p,q)}_{\infty, 0}, \alpha_i \alpha_{i+1} \ldots \alpha_n)$ and $z_{n+1} = X^{(p,q)}_{\infty, 0}$.

(ii) The solution $\nu^{\text{CR}}_{\text{op}}$ to MOES$^\infty$ always exists.

Proof: The complete proof can be found in [17]. To establish the theorem, we essentially show that the control actions of previously enabled state-feedback supervisors do not affect the control action of state-feedback supervisor $\nu^{\text{CR}}_{s, q}$, and visa versa. This implies that the MOES$^\infty$ problem is equivalent to a set of independent MOES$^0$ problems, and the optimality of the solution to MOES$^\infty$ follows from the optimality of the solutions to each of the MOES$^0$ problems.

In practice, we only need to enable state-feedback supervisors associated with a current-state estimator state $Z$ such that $Z \cap S \neq \emptyset$. Nevertheless, since we enable a new state-feedback supervisor each time we observe a new label, it seems that an infinite number of supervisors needs to be used. As it turns out, however, this scheme can be implemented with finite space complexity by taking advantage of the fact that there is a finite number of structurally different supervisors, each with a finite number of states. Next, we describe the details of this implementation.

During the operation of the system, if the current-state estimator state $Z_q$ is visited twice (for example, after observing the sequence of observations $\omega$ and $\omega(\overline{\Omega})$) then the state-feedback supervisor $\nu^{\text{CR}}_{s, q}$ associated with the current-state estimator state $Z_q$ should be enabled twice. To model this, we store a single copy of $\nu^{\text{CR}}_{s, q}$ but allow it to simultaneously lie in more than one state. Then, the control action for each $\nu^{\text{CR}}_{s, q}$ can be defined as the intersection of the control actions at each of its current states. The implementation of the supervisor in this way results in the implementation of a non-deterministic finite automaton which is relatively straightforward and requires finite memory. From this point onwards, upon observing a new label, both of the states are updated and the control action is obtained by taking the intersection of the control actions in the resulting states. In this way, we can implement two or more state-feedback supervisors that share the same structure $\nu^{\text{CR}}_{s, q}$ but may differ in their current states (and hence control actions). If the state-feedback supervisor $\nu^{\text{CR}}_{s, q}$ is re-enabled, we can use the same approach: at all times, the control action is defined as the intersection of the control actions of all possible current states of $\nu^{\text{CR}}_{s, q}$.

To keep track of the enabled state-feedback supervisors along with their current states, we define a binary state-indicator vector for each of the state-feedback supervisors in the bank. The size of this vector equals the number of states of the corresponding state-feedback supervisor and each of its elements, corresponds to a state of the state-feedback supervisor. Once a supervisor is enabled, the first element of the state-indicator vector (corresponding to the initial state of this supervisor) becomes “1”. If following the observation of an event $\alpha \in \Sigma_{\text{obs}}$, the supervisor state evolves from state $i$ to state $j$, the $j^{th}$ element in the indicator vector for the next state (initially taken to be the all zero vector) becomes “1”. If this observation also requires that the state-feedback supervisor is re-enabled, i.e., if the associated current-state estimator state is revisited, we update the indicator vector of the next state as described above and also insert a “1” in the first element of the state-indicator vector.

**Algorithm C** Consider a non-deterministic finite automaton $G = (X, \Sigma, \delta, X_0)$, with a set of observable events $\Sigma_{\text{obs}}$, a set of controllable events $\Sigma_c$, and a set of secret states $S \subseteq X$, a projection map $P$ with respect to the set of observable events $\Sigma_{\text{obs}}$, and a prefix-closed, normal and regular language $E \subseteq L(G)$, $E \neq \emptyset$, describing the legal behavior. Construct the non-deterministic finite automaton $G_E = (X_E, \Sigma_E, \delta_E, X_{0E})$ with $L(G_E) = E$, the non-deterministic finite automaton
of the solution $\nu_{op}^{\lceil \text{CNP}\rceil \infty}$ to MOES$^{\infty}$ after observing the sequence of observations $\omega = \alpha_0\alpha_1\ldots\alpha_n$ is

$$
\nu_{op}^{\lceil \text{CNP}\rceil \infty} (\omega) = \bigcup_{q=0}^{M-1} \nu_{s,q}^{\lceil \text{CR} \rceil} (I_q(\omega)).
$$

**Example 26.** Consider the DES $G$ in Figure 1-a with $\Sigma_e = \{\beta\}$ and $E = L(G)$. As mentioned in Example 8, this system is not $(\{3\}, P, \infty)$-opaque since observing $\alpha_3\beta\alpha^\ast$ reveals that secret state 3 was visited in the past. We follow Algorithm B to design a minimally restrictive supervisor which enforces infinite-step opacity. First, we need to construct the current-state estimator and the associated bank of ISEs. In Example 8, we carried this step as part of the verification process and argued that we only need to construct the two ISEs $G_{\infty,\text{obs}}^{p,3}$ and $G_{\infty,\text{obs}}^{p,4}$ associated with current-state estimator states $Z_3 = \{2, 3, 4\}$ and $Z_4 = \{0, 1, 2, 3, 4\}$. Next, following Algorithm B, we construct the state-feedback supervisor $\nu_{s,3}^{\lceil \text{CR} \rceil}$ which avoids the states in ISE $G_{\infty,\text{obs}}^{p,3}$ ($G_{\infty,\text{obs}}^{p,4}$) for which the set of starting states of the associated state mapping contains no state outside the set of secret states $\{3\}$.

It can be easily verified that the sets of starting states of all state mappings associated with ISE $G_{\infty,\text{obs}}^{p,3}$ (depicted in Figure 2-b) have states outside the set of secret states $\{3\}$ and, hence, no supervision is required (i.e., we do not need to construct $\nu_{s,3}^{\lceil \text{CR} \rceil}$). On the other hand, the set of starting states of state mapping $m_2 = \{(3, 4)\}$ associated with ISE $G_{\infty,\text{obs}}^{p,3}$ (depicted in Figure 3) only contains state 3 and, hence, violates infinite-step opacity. The supervisor $\nu_{s,3}^{\lceil \text{CR} \rceil}$ which avoids reaching that state is depicted in Figure 5-a. As a result of all of the above discussions, the bank of state-feedback supervisors only contains one supervisor $\nu_{s,3}^{\lceil \text{CR} \rceil}$. The state-indicator vector $I_3$ associated with the state-feedback supervisor $\nu_{s,3}^{\lceil \text{CR} \rceil}$ is initialized to $I_3 = (0, 0)^T$. Upon observing $\alpha$, state $Z_3$ is reached in the current-state estimator from its initial state $Z_4$ and, hence, the state-feedback supervisor $\nu_{s,3}^{\lceil \text{CR} \rceil}$ associated with state $Z_3$ becomes enabled, i.e., $I_3 = (1, 0)^T$. Upon activation of $\nu_{s,3}^{\lceil \text{CR} \rceil}$, the controllable event $\beta$ is disabled. After that, supervisor $\nu_{s,3}^{\lceil \text{CR} \rceil}$ transitions to state 1, i.e., $I_3 = (0, 1)^T$, and does not limit the behavior of the system anymore. Also, since state $Z_3$ is not reachable in the current-state estimator again, supervisor $\nu_{s,3}^{\lceil \text{CR} \rceil}$ will not be re-enabled in future. Hence, the net effect of this
supervision is the removal of the sequence of observations $\alpha\beta\alpha^*$ from the set of observations that the system could generate. (Note that if at the very beginning $\beta$ is observed, then no supervisor will ever be enabled.)

\textbf{Remark 27.} In order to implement the solution to MOES$^\infty$ using Algorithm B, one needs to store (i) the bank of state-feedback supervisors $V^\text{CSR}_{s,q}$, $0 \leq q \leq M - 1$, and (ii) the bank of state-indicator vectors $I_q$, $0 \leq q \leq M - 1$, associated with each state-feedback supervisor. Here, $M$ is the number of states of the current-state estimator associated with DES $G$. If $N$ is the number of states of $G$, then the bank of state-feedback supervisors has $O(2^N \times 2^{N^2})$ state-space complexity (each state-feedback supervisor is constructed from an ISE-like automaton (which has $O(2^{N^2})$ state-space complexity) and there are at most $O(2^N)$ such state-feedback supervisors (associated with each state of the current-state estimator)). Each state-indicator vector is a vector of 0s and 1s and contains at most $2^{N^2}$ elements (which is the maximum number of states of a state-feedback supervisor). Therefore, storing the bank of state-indicator vectors requires $O(2^N \times 2^{N^2})$ bits. The exponential complexity of the solution to MOES$^\infty$ is not surprising since algorithms that implement minimally restrictive supervisors for non-deterministic automata with partial observations (e.g., to enforce diagnosability [19]) typically have similar complexity [19].

\textbf{Remark 28.} As mentioned earlier, in the framework considered in [7], [8], the set of strings $L(G)$ in the system is partitioned into secret strings $K$ and non-secret strings $L(G) - K$, and the system is opaque if $P(K) \subseteq P(L(G) - K)$. Unlike the notion of initial-state opacity which can be formulated using the framework of [7], [8] (refer to Remark 23), the notion of infinite-step opacity cannot be easily translated to a version of language-based opacity as studied in [7], [8]. This is due to the fact that in the framework we consider here, each string in the system can be treated both as secret (when it passes through the set of secret states) and non-secret (when it does not pass through the set of secret states); hence, we cannot partition the strings in the system into secret and non-secret ones.

\textbf{Remark 29.} The notion of non-interference is defined assuming that the set of events $\Sigma$ can be partitioned into a finite number of security levels [10]. If we consider the case when there are only two security levels, denoting these two partitions by public events $\Sigma_{pu}$ and private events $\Sigma_{pr}$ (so that $\Sigma = \Sigma_{pu} \cup \Sigma_{pr}$ and $\Sigma_{pu} \cap \Sigma_{pr} = \emptyset$), non-interference requires that a user who only observes public events cannot determine the occurrence of private events [10]. The notion of non-interference for this scenario can be translated to an instance of 0-step opacity [4] as follows. We first obtain the non-deterministic automaton $\hat{G} = G \times H$, where the deterministic automaton $H$ is depicted in Figure 5-b. If we define $S = \{(x, Pr) \mid x \in X\}$, then it is not hard to argue that $\hat{G}$ is non-interferent if and only if $\hat{G}$ is $(S, P, 0)$-opaque, where projection map $P$ is with respect to the set $\Sigma_{pu}$ of public events. Since the results of this paper can be easily extended to design minimally restrictive supervisors that enforce 0-step opacity (as briefly discussed in Section V), the results in [10] can be generated using the framework of this paper. However, it is not clear if one can formulate the notions of initial-state and infinite-step opacity in the framework of [10].

\section*{VI. Conclusions}

In this paper, we consider the problem of designing feasible supervisors that enforce state-based notions of opacity while limiting the behavior of an underlying system (modeled as a non-deterministic finite automaton) to a subset behavior, called legal behavior and described by a prefix-closed language $E$. More specifically, we consider two notions of opacity: initial-state opacity, which requires opacity of the membership of the initial state of the system to the set of secret states, and infinite-step opacity, which requires opacity of the membership of the system state to the set of secret states at any point along the observation process. Under the assumption that all controllable events are observable ($\Sigma_c \subseteq \Sigma_{obs}$), we characterize the set of solutions to the initial-state opacity problem as the set of sublanguages of $E$ that are controllable, normal, and opaque. We show that there always exists a minimally restrictive solution to this problem and propose a method to find the supremal of such languages (which is the solution of our minimally restrictive supervisory control problem). We implement the solution using state estimators and show how to utilize a bank of such controllers to enforce infinite-step opacity.
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. The control problem can then be connected to the design of control policies for stochastic systems under suitable optimality criteria for probabilistic opacity.

REFERENCES


Aonooshiran Saboori received the B.Sc degree in Electrical and Computer Engineering in 2003 from Sharif University of Technology, Tehran, the M.Sc degree in Electrical and Computer Engineering in 2005 from Concordia University, Montreal, QC, and the Ph.D. degree in Electrical and Computer Engineering in 2010 from the University of Illinois at Urbana-Champaign, IL.

In 2010, he joined the Windows Security group at Microsoft as a program manager. His research interests include modeling, analysis, and control of discrete event systems with applications to network security, sensor networks, and manufacturing systems.

Christoforos N. Hadjicostis (M’99, SM’05) received S.B. degrees in Electrical Engineering, in Computer Science and Engineering, and in Mathematics, the M.Eng. degree in Electrical Engineering and Computer Science in 1995, and the Ph.D. degree in Electrical Engineering and Computer Science in 1999, all from the Massachusetts Institute of Technology, Cambridge, MA. In 1999 he joined the Faculty at the University of Illinois at Urbana-Champaign where he served as assistant and then associate professor with the Department of Electrical and Computer Engineering, the Coordinated Science Laboratory, and the Information Trust Institute. Since 2007, Dr. Hadjicostis has been with the Department of Electrical and Computer Engineering at the University of Cyprus. His research focuses on fault diagnosis and tolerance in distributed dynamic systems; error control coding; monitoring, diagnosis and control of large-scale discrete event systems; and applications to network security, anomaly detection, energy distribution systems, medical diagnosis, biosequencing, and genetic regulatory models.