Opacity of Discrete Event Systems with Active Intruder

Alireza Partovi Student Member, IEEE, Tacho Jung, Member, IEEE, and Hai Lin, Senior Member, IEEE

Abstract—Opacity is a security property formalizing the information leakage of a system to an external observer, namely an intruder. The conventional opacity that has been studied in the Discrete Event System (DES) literature usually assumes passive intruders, who only observe the behavior of the system. However, in many cybersecurity concerns, such as web service, active intruders, who are capable of influencing the system’s behavior beyond passive observations, need to be considered and defended against. We are therefore motivated to extend the opacity notions to handle active intruders. For this, we model the system as a non-deterministic finite-state transducer. It is assumed that the intruder has a full knowledge of the system structure and is capable of interacting with the system by injecting different inputs and observing its responses. In this setup, we first introduce reactive current-state opacity (RCSO) notion characterizing a property that the system does not leak its secret state regardless of how the intruder manipulates the system behavior. We furthermore extend this notion to reactive initial-state opacity notions, and study the relationship among them. It turns out that all the proposed reactive opacity notions are equivalent to RCSO. We therefore focus on RCSO and study its verification problem. It is shown that the RCSO can be verified by constructing an observer automaton. Illustrative examples are provided throughout the paper to demonstrate the key definition and the effectiveness of the proposed opacity verification approach.

I. INTRODUCTION

Cybersecurity is increasingly becoming a great concern as networks of embedded-systems and computers are integrated into almost all aspects of our daily life and society. Exchanging confidential information over these networks is crucial in many applications, ranging from smart phones and home automation to banking services. This raises a serious concern on the vulnerability of these systems.

Many efforts have been made to develop reliable and secure systems that led to various notions of security/privacy. One class of security/privacy notations is related to Information flow from the system to an external observer [1]. Opacity is a type of information-flow property that characterizes whether the system’s secret information can be inferred by an external observer termed intruder with potentially malicious intentions [2]. It is usually assumed that the intruder knows the system’s structure but has only partial observation over its behavior [3]. The system is considered to be opaque if the intruder is not able to unambiguously determine the system secrets from its observations.

In recent years, opacity has been extensively studied in the discrete event system (DES) literature, and different notions of opacity have been proposed, including current-state opacity [4], language-based opacity [2], initial-state opacity [5], $K$–step, and infinite-step opacity [6]. Interested readers may refer to [3] for a comprehensive review on various notions of opacity.

It is worthy pointing out that the intruder model considered in these methods is a passive observer who is only able to partially observe the system behavior. However, many real-world systems are interacting with malicious and hostile environments, whose capability is beyond a passive observation. A system’s malicious environment can act as an active intruder, who strategically injects a certain input to the system and observes the system’s response to infer its secret. For instance, web browsers and client-side web applications are typical cases of such systems since they interact with remote and possibly untrusted clients that raise a serious concern about the privacy of local users’ data [7].

In this paper, we aim at extending the opacity notion in the presence of an active intruder. In particular, who is capable of manipulating the system’s input and partially observing the system output. This setup naturally models reactive systems [8], such as interactive programs [9] and web services [7], where input provided by the environment (possibly intruder) and the output of the system is exchanged continuously throughout the indefinite execution of the system.

Toward this aim, we introduce reactive current-state opacity (RCSO) characterizing the active intruder’s ability in manipulating the system’s input to certainly determine if the system’s current-state is a secret state. We furthermore extend this notion to reactive language-based opacity and reactive initial-state opacity. Reactive language-based opacity requires the secret behavior of the system to be indistinguishable from a non-secret one. Reactive initial-state opacity notions ensure the active intruder cannot unambiguously determine if the system starts from a secret initial-state. Upon these opacity notions, we present their relationship, the feasibility of each notion, and a procedure to transform one to the other. It turns out that all the proposed reactive opacity notions are equivalent to RCSO. We therefore focus on RCSO, and we study its verification problem.

Formal verification of current-state opacity is addressed in [10] and is further extended to other notions of opacity in [4], [5]. In analogs to verification of opacity with the passive intruder, here we propose to construct an observer automata. Given the intruder choice of input and the system response (the observable output event), the observer states capture the estimated current-state of the system. Hence, the
RCSO verification problem can be reduced to finding the observer states that include a singleton of the secret states.

The contribution of this paper can be summarized as follows. (i) Consider a new intruder model who has the capability of injecting input into the system; (ii) associated with the new intruder model, we introduce a new class of opacity definitions including the reactive current-state, reactive initial-state, and reactive language-based opacity notions and studies the relationship among them; (iii) provide necessary and sufficient conditions for verification of reactive current-state opacity.

II. RELATED NOTATIONS

In this section, we review some preliminary notations that will be used throughout the paper. For a given finite set (alphabet) of events $$\Sigma$$, a finite word $$w = \sigma_1 \sigma_2 \ldots \sigma_n$$, $$n \geq 1$$, is a finite sequence of elements in $$\Sigma$$, for all $$\sigma_i \in \Sigma$$, and $$1 \leq i \leq n$$. We denote the length of $$w$$ by $$|w|$$. Let $$w$$ and $$u$$ be finite words, $$w \cdot u$$ is their concatenation. The notation $$2^\Sigma$$ refers to the power set of $$\Sigma$$, that is, the set of all subsets of $$\Sigma$$. A set difference is $$\Sigma - A = \{x \mid x \in \Sigma, x \notin A\}$$. The free monoid $$\Sigma^*$$ generated by $$\Sigma$$ is the set of all finite sequences $$\sigma_1 \sigma_2 \ldots \sigma_n$$, including the empty sequence denoted by $$\epsilon$$. A subset of $$\Sigma^*$$ is called a language over $$\Sigma$$. The prefix-closure of a language $$L \subseteq \Sigma^*$$, denoted as $$\overline{L}$$, is the set of all prefixes of words in $$L$$, i.e., $$\overline{L} = \{s \in \Sigma^* \mid \exists t \in L \}$$; $$\overline{L}$$ is said to be prefix-closed if $$\overline{L} = L$$. Let’s consider alphabet sets $$X$$, $$Y$$, and their set product $$\Sigma_{XY} = X \times Y$$. A relation $$R$$ over sets $$X$$ and $$Y$$ is a subset of the Cartesian product $$X \times Y$$. A regular (or rational) relation over the alphabets $$X$$ and $$Y$$ is formed from a finite combination of the following rules: 1: $$(x, y) \in (X \cup \{\epsilon\}) \times (Y \cup \{\epsilon\})$$, 2: $$\varnothing$$ is a regular relation, and 3: If $$R_1$$, $$R_2$$ are regular relations, then so are $$R_1 \cup R_2$$, $$R_1 \cap R_2$$, and $$R_1^*$$.

Projection function to sets $$X$$ and $$Y$$ are respectively denoted as $$P_X = \Sigma_{XY} \rightarrow X^*$$, $$P_Y = \Sigma_{XY} \rightarrow Y^*$$, and inductively are defined by $$P_X(\epsilon, x) = x$$, and $$\forall w \in \Sigma_{XY}$$, $$(x, y) \in \Sigma_{XY}$$, we have $$P_X(w \cdot (x, y)) = P_X(w \cdot x)$$, and $$P_Y(w \cdot (x, y)) = P_Y(w) \cdot y$$.

A non-deterministic finite state automata (NFA) $$A = (Q, \Delta, Q_0, T_a)$$ is a 4-tuple composed of finite state $$Q$$, a finite set of events $$\Delta$$, a partial state transition function $$T_a : Q \times \Delta \rightarrow 2^Q$$, and the set of initial states $$Q_0$$. The transition function $$T_a$$ can be extended to word in a standard recursive manner. The behavior of NFA $$A$$ is captured by $$L(A) = \{s \in \Delta^* \mid \exists q_0 \in Q_0 \text{ s.t. } T_a(q_0, s) \neq \varnothing\}$$, and for a given initial state $$q_0 \in Q_0$$ is $$L(A, q_0) = \{s \in \Delta^* \mid T_a(q_0, s) \neq \varnothing\}$$. $$A$$ is called deterministic finite automata (DFA) if for any $$q \in Q$$ and $$\delta \in \Delta$$ that $$T(q, \delta)$$ is defined, $$|T_a(q, \delta)| = 1$$.

III. OPEN DISCRETE EVENT SYSTEM

The finite-state transducers capture transformation of data that is realized by processing inputs and producing outputs using finite memory [11]. We use non-deterministic finite-state transducer (NFT) to characterize the interaction between the system and its environment. Throughout this paper, we refer to NFT as an open DES to emphasize a system model which receives input from an active intruder.

**Definition 1 (Non-deterministic Finite-State Transducer):** The nondeterministic finite-state transducer is defined by $$G = (Q, X, \Delta, Q_0, T, \lambda)$$, where $$Q$$ is the finite set of states, $$X$$ is finite set of external events, $$\Delta = \Delta_o \cup \Delta_u$$, is the finite set of output events which is partitioned to two disjoint sets of observable output events $$\Delta_o$$ and unobservable output events $$\Delta_u = Q_0$$ is the set of initial states. The state transition function is $$T : Q \times X \rightarrow 2^Q$$, and $$\lambda : Q \times X \rightarrow 2^\Sigma$$ is the output function, where $$X = X \cup \{\epsilon\}$$ and $$\Delta = \Delta \cup \{\epsilon\}$$.

The notation $$T(q, x)$$ means that $$T(q, x)$$ is defined for $$x \in X$$ and state $$q \in Q$$. The extension of $$T$$ to words is denoted as $$T^* : Q \times X^* \rightarrow 2^Q$$ and can be defined recursively for all $$q \in Q$$ as $$T^*(q, w) = q$$ if $$w = \epsilon$$, and $$T^*(q, w) = \bigcup_{q' \in T(q, x)} T^*(q', v)$$ if $$w = x \cdot v, x \in X$$, and $$v \in X^*$$ [12]. Here, $$T(q, \epsilon) = q$$ for each $$q \in Q$$, indicates that if the input is the empty word, we will remain at the current state. The extension of output function to words also is denoted as $$\lambda^* : Q \times X^* \rightarrow 2^\Sigma$$, and it can be defined as follows. Given any $$w \in X^*$$, and $$s \in \Delta^*$$, we have $$s \in \lambda^*(q, w)$$ for some $$q \in Q$$, if and only if, either $$w = s = \epsilon$$, or $$w = x \cdot w', s = \delta \cdot s'$$ for some $$x \in X$$, and $$\delta \in \Delta$$, and there exists a state $$q' \in Q$$ such that $$q' = T(q, x)$$, $$\delta = \lambda(q, x)$$, and $$s' \in \lambda^*(q', w')$$. The recognized language of $$G$$ is $$L(G, Q_0) = \{w \in X^* \mid \exists q_0 \in Q_0 \text{ s.t. } T(q_0, w)\}$$. Throughout the paper, we use $$T$$ as a shorthand for $$T^*$$, $$\lambda$$, and $$L(G, Q_0)$$.

Given an input word $$w \in L(G)$$, the output word will not be uniquely determined, due to the non-determinism of the transition and output functions. For each $$q_0 \in Q$$ and $$w \in L(G, q_0)$$, a set $$O(w, q_0)$$ of possible output words is defined inductively as follows:

- $$O(\epsilon, q_0) = \{\epsilon\}$$,
- $$\forall w \in L(G, q_0), \forall x \in X$$, such that $$w \cdot x \in L(G, q_0)$$:
- $$O(w \cdot x, q_0) = \{s \cdot \delta \in \Delta^* \mid s \in O(w, q_0) \text{ and } \delta \in \bigcup_{q' \in T(q_0, x)} \lambda(q', x)\}$$.

We denote $$O(w) = \bigcup_{q_0 \in Q} O(w, q_0)$$. The set of all possible output words in $$G$$ is denoted by $$O(L(G))$$, that is, $$O(L(G)) = \bigcup_{q_0 \in Q} O(L(G, q_0)) \subseteq \Delta^*$$.

**Example 1:** Consider the open DES shown in Figure 1 where $$\Delta = \{\delta_1, \delta_2, a, b\}$$, $$X = \{x_1, x_2\}$$, and the initial state is $$Q_0 = \{0\}$$. An edge in the model is in the form of $$x/Y$$, where $$x \in X$$, represents the input event, and and $$Y \in \Delta$$. 

---

**Fig. 1.** An example of open DES G. Note that $$\epsilon \in \lambda(q, x)$$ for all $$q \in Q$$, and $$x \in X_e$$. We removed the $$\epsilon$$ input transitions for clarity of the figures.
denotes the set of possible output events. Multiple labels over an edge indicates multiple enabled transitions. For instance, for \( x_1 x_1 \in \mathcal{L}(G) \), we have \( O(x_1 x_1) = \{ \delta_1 \delta_2, \delta_2 \delta_2 \} \), that is, two output words, \( \delta_1 \delta_2 \), and \( \delta_2 \delta_2 \) are possible.

If there are marked states, we define open DES as \( G = (Q, X, \Delta, Q_0, T, \lambda, F) \), where \( F \subseteq Q \) are the marked states. The input-output language of \( G \), denoted as \( \mathcal{L}_i(G) \), is defined by \( \mathcal{L}_i(G) = \{ (w, s) \mid (X \times \Delta)^* \} \), \( \bar{\lambda}(q_0, w) \in Q_0 \), s.t. \( T(q_0, w) \), and \( s \in \lambda(q_0, w) \), and its input-output marked language is given by \( \mathcal{L}_m(G) = \{ (w, s) \mid (X \times \Delta)^* \} \), \( \bar{\lambda}(q_0, w) \in Q_0 \), s.t. \( T(q_0, w) \in F \neq \emptyset \), and \( s \in \lambda(q_0, w) \). The input-output languages of \( G \) is a regular relation over the set \( (X \cup \{ \} \times (\Delta \cup \{ \} \) that can be conveniently recognized by an non-deterministic finite-state transducer [13].

The accessible part of an NFT \( G = (Q, X, \Delta, Q_0, T, \lambda, F) \) is denoted by \( \operatorname{Ac}(G) \) and is obtained by removing the states that cannot be reached from any initial state \( q_0 \in Q_0 \) in finite number of steps. The coaccessible part of \( G \), denoted by \( \operatorname{CoAc}(G) \) is an NFT obtained by deleting the states that cannot reach to the marked states \( F \). The trim operation, denoted by \( \operatorname{Trim} \), transforms \( G \) to another NFT as a part of \( G \) that is both accessible and coaccessible, formally \( \operatorname{Trim}(G) = \operatorname{Ac}(\operatorname{CoAc}(G)) = \operatorname{CoAc}(\operatorname{Ac}(G)) \) [14]. Similarly, for an NFA \( A \), we can define \( \operatorname{Trim}(A) \), \( \operatorname{Ac}(A) \), and \( \operatorname{CoAc}(A) \).

IV.Opacity Of Discrete-Event Systems

Opacity is characterized by the system’s secret and the intruder’s observation mapping over the system’s executions. The system is opaque, if for any execution run that contains secret, there exists another non-secret run which is observably equivalent. In the formalism of opacity, the intruder is considered as an observer who has full knowledge of the system structure but has a partial observability over it. Typically, the intruder’s partial observability is modeled by a natural projection function. The natural projection is \( P : \Delta^* \rightarrow \Delta_s^* \), and for any \( s \in \Delta_s \), and \( \delta \in \Delta \), it is defined recursively by \( P(\epsilon) = \epsilon \), and \( P(s \cdot \delta) = P(s) \cdot \delta \) if \( \delta \in \Delta_o \) and otherwise \( P(s \cdot \delta) = P(s) \).

The system secret information or behavior can be represented in different ways, such as secret states and languages. In the conventional opacity of DESs with passive intruder, various opacity notions for different representation of secret have been introduced including but not limited to current-state, language-based, and initial-state opacity [3].

A. Current-State Opacity

Here, we first discuss the current-state opacity (CSO) definition when the intruder is just a passive observer; and later, we will show how an active intruder can force a current-state opaque system to expose its secret states.

**Definition 2 (Current-State Opacity):** Given a non-deterministic finite-state automata \( A = (Q, \Delta, Q_0, T_A) \), and a passive intruder with projection function \( P \), a set of secret state \( Q_s \subseteq Q \), the system \( A \) is current-state opaque if \( \forall q_0 \in Q_0 \) and \( \forall s \in \mathcal{L}(A, q_0) \) such that \( T_A(q_0, s) \subseteq Q_s \), there exists \( q'_0 \in Q_0 \) and \( \exists s' \in \mathcal{L}(A, q'_0) \), such that \( T_A(q'_0, s') \subseteq \{ Q - Q_s \} \) and \( P(s) = P(s') \).

Intuitively, when the intruder can only observe the system outputs with projection \( P \), \( A \) is current-state opaque if for every word \( s \in \mathcal{L}(A) \) leading to a secret state in \( Q_s \), there exists at least another word \( s' \in \mathcal{L}(A) \) that leads to non-secret states \( \{ Q - Q_s \} \) whose projection is the same. Thus, the intruder can never determine that the system’s current state is in \( Q_s \). One can check whether the system \( A \) with a passive intruder is current-state opaque by constructing a current-state estimator (observer) and by verifying that no (nonempty) current-state estimate lies entirely within the set of secret states \( Q_s \) [15].

**Example 2:** Consider the open DES \( G \) depicted in Figure 1 with \( \Delta_o = \{ \delta_1, \delta_2, a \} \), \( \Delta_{uo} = \{ b \} \), and \( Q_s = \{ 3 \} \). We first assume the intruder is passive and can only observe the observable outputs through projection function \( P \). In order to evaluate CSO on \( G \), we can associate a NFA \( A \) with the open DES \( G \). Let’s consider the NFA \( A_G = (Q, \Delta, Q_0, T_G) \), where the transition function \( T_G \), for any \( q, a \in Q \), and \( \delta \in \Delta \), is defined as \( q' \in T_G(q, \delta) \), if there exists \( x \in X \) such that \( q' \in T(q, x) \) and \( \delta \in \lambda(q, x) \); otherwise \( T_G(q, \delta) \) is not defined. We can construct an observer automata to check if \( A_G \) is current-state opaque with respect to \( P \), and \( Q_s \). The observer is shown in Figure 2. The observer shows the secret state \( \{ 3 \} \) never lies entirely on single state of the observer, and hence, \( A \) is current-state opaque with respect to \( Q_s \) and \( P \). However, if the intruder is capable of providing a certain input word to the system and observe the system’s output through \( P \), she can infer when the system is in the secret state. Specifically, consider the input word \( w = x_1 x_2 x_3 \) that drives the system to land on one of the states \( \{ 2, 3 \} \), and here, if the active intruder chooses \( x_2 \), i.e., \( w = x_2 \) and observes \( a \), she can infer the current-state of the system is certainly at the secret state \( \{ 3 \} \). However, if \( a \) is an unobservable event, the active intruder with the same input word \( x_1 x_2 x_1 x_2 x_2 \), cannot determine whether the system is at \( \{ 3 \} \) or \( \{ 2 \} \).

As Example 2 illustrates, an active intruder can force the open DES \( G \) to expose his secret-state. We, therefore, need a new current-state opacity notion that captures this active intruder ability. In particular, we consider an active intruder who has full knowledge of the open DES model; and is capable of injecting input to the system and (partially) observing the system output.

To evaluate an open DES current-state opacity, we can construct a current-state estimator that tracks the active intruder estimated states. Given an input word accepted by the system \( w \in \mathcal{L}(G) \), and an observed word \( \alpha \in P(O(\mathcal{L}(G))) \),

![Fig. 2. Current-state estimator of the passive intruder for the open DES in Figure 1]
the current-state estimator is defined by:

$$Q^G(w, \alpha) = \{ q \in Q \ | \exists q_0 \in Q, q \in T(q_0, w), \text{ and } \exists s \in O(w, q_0), \text{ s.t. } P(s) = \alpha \}. $$

The current-state estimator $\hat{Q}^G(w, \alpha)$ essentially characterizes a set of states which the open DES lands on as a result of the input word $w$, and meanwhile it produces the observable sequences $\alpha$. We also define the current-state estimator for a given initial state $q_0 \in Q_0$, as $Q^G(q_0, \alpha) = \{ q \in Q \ | \ q \in T(q_0, w), \text{ and } \exists s \in O(w, q_0), \text{ s.t. } P(s) = \alpha \}$. We use $Q(w, \alpha)$ instead of $Q^G(w, \alpha)$, and $Q(w, \alpha)$ for $\hat{Q}^G(w, \alpha)$, when it is clear from the context. Upon this current-state estimator, we define the reactive current-state opacity in the following.

**Definition 3 (Reactive Current-State Opacity):** Given an open DES $G = (Q, X, \Delta, Q_0, T, \lambda)$, projection function $P$, and the set of secret states $Q_s \subseteq Q$, the system is reactive current-state opaque (RCS-opaque) if for any $w \in L(G)$ there exists $q_0 \in Q_0$ such that:

- $T(q_0, w) \cap \{Q - Q_s\} \neq \emptyset$,
- $\forall t \in P(O(w, q_0))$, we have $Q_{q_0}(w, t) \cap \{Q - Q_s\} \neq \emptyset$.

Intuitively, the open DES $G$ is RCS-opaque, if with any input word $w$ that is recognized by $G$, i.e., $w \in L(G)$, i) there exists an initial state $q_0 \in Q_0$ such that the system with $w$ does not land entirely at the secret states, i.e., $T(q_0, w) \cap \{Q - Q_s\} \neq \emptyset$; and ii) for any possible observable output word associated with the input, $t \in P(O(w, q_0))$, we have $Q_{q_0}(w, t) \cap \{Q - Q_s\} \neq \emptyset$, that is, the intruder cannot use the observed output events to resolve the non-determinism of the transition function $T(q_0, w)$ to infer the current secret state of the system.

**Remark 1:** In the definition of RCSO, the input word $w$, is not required to be restricted to the recognized words by the open DES $G$, $w \in L(G)$, and it can be any $w \in X^*$. However, clearly $G$ does not accept any $w \in \{X^* - L(G)\}$, and hence, it does not reveal any secret.

**Example 3:** Consider the system $G$ in Figure 1 with secret state set $Q_s = \{2\}$. In this case, $G$ is not RCSO since the intruder with input word $w = x_2$ and regardless of the observed output events, can ensure the system current-state is $\{2\}$. However, if $Q_s = \{3\}$, the system with any $w \in L(G)$, does not proceed solely to $Q_s$, and therefore, the intruder potenially can use the observed output events to infer the secret state from the system’s possible current-states. For instance, with $x_1x_1x_2$, the possible current-states of the system are $\{2, 3\}$, and if the observed output word is $t \cdot a$, where $t$ is any $t \in O(x_1x_1)$, the intruder is able to certainly infer the current state of $G$ is the secret state $\{3\}$, that indicates $G$ is not RCS-opaque.

**Remark 2:** The proposed RCSO notion with an active intruder is a generalization of CSO notion with the passive intruder. As it is illustrated in Example 2 if we consider open DES with a passive intruder who has a partial observation on the system’s output, the proposed RCSO can capture the CSO notion.

**B. Other Opacity Notions**

Other notions of opacity can be extended to the open DESs with an active intruder. In this paper, we introduce reactive language-based and reactive initial-state opacity notions. The reactive language-based opacity (RLBO) characterizes a secret run of the system that should be protected against an active intruder.

**Definition 4 (Reactive Language-Based Opacity):** Given an open DES $G = (Q, X, \Delta, Q_0, T, \lambda)$, projection function $P$, and secret output language $O_s \subseteq O(L(G))$, and non-secret output language $O_{ns} \subseteq O(L(G))$. $G$ is reactive language-based opaque, if for all $q_0 \in Q_0$, and any $w \in L(G, q_0)$ that $O(w, q_0) \cap O_s \neq \emptyset$, there exists $q_0' \in Q_0$ such that:

- $O(w, q_0') \cap O_{ns} \neq \emptyset$,
- $\forall t \in (O(w, q_0) \cap O_s), \exists t' \in (O(w, q_0') \cap O_{ns})$ such that $P(t) = P(t')$.

Intuitively, $G$ is reactive language-based opaque with respect to the secret output language $O_s$, non-secret output language $O_{ns}$, and the projection function $P$, if for any input word $w \in L(G, q_0)$ that generates secret output word, $O(w, q_0) \cap O_s \neq \emptyset$, there exists an initial state $q_0' \in Q_0$, such that the same input word from the intruder can be associated with a non-secret output word, $O(w, q_0') \cap O_{ns} \neq \emptyset$, and additionally, for any secret output word $t \in O(w, q_0) \cap O_s$ there exists a non-secret output word $t' \in O(w, q_0') \cap O_{ns}$, such that they have the same observation $P(t) = P(t')$.

Initial-state opacity is another notion of opacity defined over the system secret initial states. For open DESs, reactive initial-state opacity (RISO) can be defined as follows.

**Definition 5:** (Reactive Initial-State Opacity) Given an open DES $G = (Q, X, \Delta, Q_0, T, \lambda)$, projection function $P$, and secret initial-state set $Q_0^0 \subseteq Q_0$, and non-secret initial-state set $Q_{ns}^0 \subseteq Q_0$, $G$ is reactive initial-state opaque, if $\forall q_0 \in Q_0^0$ and any input words $w \in L(G)$ with any $t \in O(w, q_0)$, there exists a non-secret initial-state $q_0' \in Q_{ns}^0$ and $t' \in O(w, q_0')$ such that $P(t) = P(t')$.

An open DES $G$ is reactive initial-state opaque with respect to the secret initial-state set $Q_0^0$, non-secret initial-state set $Q_{ns}^0$, and the projection function $P$, if for any secret initial-state $q_0 \in Q_0^0$, and any input word $w \in L(G)$, that generates an output word $t$, i.e., $t \in O(w, q_0)$, there exists a non-secret initial state $q_0' \in Q_{ns}^0$, and an output word $t' \in O(w, q_0')$, associated with $w$ and $q_0'$ such that, $t$ and $t'$ have the same observation, i.e., $P(t) = P(t')$.

Similar to the opacity notions with a passive intruder [16], there is a relationship between the proposed reactive opacity notions. We call a problem of checking if a given open DES satisfies the RCSO conditions, a RCSO problem. Similarly, in the sequel, we use the terms RLBO and RISO problems. We mainly follow the idea proposed in [16] to transform the reactive opacity problems to each other.

**Proposition 1:** A RLBO problem can be converted to an equivalent RCSO problem.

**Proof:** Construct an NFT $G_s = (S_s, X, \Delta, T_s, S_{s0}, \lambda_s, F_s)$ such that $L_{io,m}(G_s) =$
\{(w, s) \in (X \times \Delta)^* \mid w \in \mathcal{L}(G) \text{ and } s \in O_s\}, \text{ and an NFT } G_{ns} = (S_{ns}, X, \Delta, T_{ns}, S_{ns0}, \lambda_{ns}, F_{ns}) \text{ that accepts } \mathcal{L}_{io,m}(G_{ns}) = \{(w, s) \in (X \times \Delta)^* \mid w \in \mathcal{L}(G) \text{ and } s \in O_{ns}\}. \text{ Then consider } G_c \text{ and } G_{ns} \text{ as single NFT by constructing } G_c = (S_c \cup S_{ns}, X, \Delta, T_c \cup T_{ns}, S_{c0} \cup S_{ns0}, \lambda_c \cup \lambda_{ns}, F_c \cup F_{ns}) \text{, and define the secret and non-secret state sets respectively by } Q_s = F_s \text{ and } Q_{ns} = F_{ns}. \text{ Therefore, for any } q_0 \in Q_0, w \in \mathcal{L}(G, q_0) \text{ and } t \in O(w, q_0) \subseteq O_s, \text{ there exist } s_0 \in (S_{ns0} \cup S_{s0}) \text{ and } \rho \in \mathcal{L}_{io,m}(G_c, s_0) \text{ with } P_X(\rho) = w \text{ and } P_{\Delta}(\rho) = t, \text{ such that } Q^\omega_{G_{ns}}(w, t) \subseteq Q_s; \text{ and if } 3q'_0 \in Q_0 \text{ and } t \in O(w, q'_0) \subseteq O_{ns}, \text{ indicating } G \text{ is reactive language-based opaque, we have } s_0' \in (S_{ns0} \cup S_{s0}) \text{ and } \rho' \in \mathcal{L}_{io,m}(G_c, s'_0) \text{ with } P_X(\rho') = w \text{ and } P_{\Delta}(\rho') = t', \text{ such that } Q^\omega_{G_{ns}}(w, t') \subseteq Q_{ns}, \text{ which implies } G_c \text{ is RCS-opaque.} \hfill \blacksquare

The other direction of this transformation is also possible. A RCSO problem can be converted to an equivalent RLBO problem.

**Proposition 2:** A RCSO problem can be converted to an equivalent RLBO problem.

**Proof:** Given an RCSO problem with \( G = (Q, X, \Delta, Q_0, T, \lambda) \), secret states \( Q_s \subseteq Q \), and non-secret states set \( Q_{ns} \subseteq Q \). Construct an NFT with \( Q_0 \) as the marked states, defined as \( G_s = \text{Trim}(Q, X, \Delta, T, Q_0, \lambda, Q_s) \), and another NFT with \( Q_{ns} \) as the marked states, given by \( G_{ns} = \text{Trim}(Q, X, \Delta, T, Q_0, \lambda, Q_{ns}) \). Then define the secret and non-secret output language respectively by \( O_s = P_{\Delta}(\mathcal{L}_{io,m}(G_s)) \) and \( O_{ns} = P_{\Delta}(\mathcal{L}_{io,m}(G_{ns})) \).

The RISO is related to the RLBO. Proposition 3 and 4 establish this relationship.

**Proposition 3:** A RISO problem can be converted to an equivalent RLBO problem.

**Proof:** Given open RISO problem with \( G = (Q, X, \Delta, Q_0, T, \lambda) \), secret initial-state set \( Q^0_s \subset Q_0 \), and non-secret initial-state set \( Q^0_{ns} \subset Q_0 \), construct an NFT by trimming \( G \) to only the secret initial-state set \( Q^0_s \subset Q_0 \), construct another NFT with \( Q^0_{ns} \) as initial-state set, \( G_{ns} = \text{Trim}(Q, X, \Delta, T, Q^0_s, \lambda) \), and similarly construct another NFT with \( Q^0_{ns} \) as initial-state set, \( G_{ns} = \text{Trim}(Q, X, \Delta, T, Q^0_s, \lambda) \). Then combine \( G_s \) and \( G_{ns} \) as \( G_0 = \text{Trim}(Q, X, \Delta, T, Q^0_s, \lambda) \), and define the secret and non-secret output languages respectively by \( O_s = \mathcal{L}(G, Q^0_s) \), and \( O_{ns} = \mathcal{L}(G, Q^0_{ns}) \).

The other direction of this transformation does not always hold. A RLBO problem can be transformed to an equivalent RISO only if \( O_s \) and \( O_{ns} \) are prefix-closed.

**Proposition 4:** Given a RLBO problem with prefix-closed \( O_s \) and \( O_{ns} \), there exists an equivalent RISO problem.

**Proof:** Given an RLBO problem with the open DES \( G = (Q, X, \Delta, T, Q_0, \lambda) \), and prefix-closed secret output language \( O_s \subset \mathcal{L}(G) \), and prefix-closed non-secret output language \( O_{ns} \subset \mathcal{L}(G) \). Construct an NFT \( G_s = (S_s, X, \Delta, T_s, S_{s0}, \lambda_s) \) such that \( \mathcal{L}_o(G_s) = \{ \rho \in (X \times \Delta)^* \mid P_X(\rho) \in \mathcal{L}(G) \text{ and } P_{\Delta}(\rho) \in O_s\} \), and an NFT \( G_{ns} = (S_{ns}, X, \Delta, T_{ns}, S_{ns0}, \lambda_{ns}) \) that accepts \( \mathcal{L}_o(G_{ns}) = \{ \rho \in (X \times \Delta)^* \mid P_X(\rho) \in \mathcal{L}(G) \text{ and } P_{\Delta}(\rho) \in O_{ns}\}. \) Then consider \( G_s \) and \( G_{ns} \) as single NFT by constructing \( G_c = (S_c \cup S_{ns}, X, \Delta, T_c \cup T_{ns}, S_{c0} \cup S_{ns0}, \lambda_c \cup \lambda_{ns}, F_c \cup F_{ns}) \), and define the secret and non-secret initial-state sets respectively as \( Q^0_s = S_{s0} \text{ and } Q^0_{ns} = S_{ns0} \).

**Remark 3:** It is shown that the proposed RCSO and RLBO are equivalent properties for \( G \). The RISO can be transformed to a RLBO property, however, the reverse of this transformation (RLBO to RISO), only holds for prefix-closed secret and non-secret languages. Therefore, if the prefix-closed conditions hold, RISO is also an equivalent property to RCSO. Figure 3 illustrates this relation.

**V. RCSO VERIFICATION**

In this section, we present the verification of RCSO notion for open DESs. Similar to current-state opacity with a passive intruder [4], we can construct an observer automata to verify if an open DES is RCS-opaque. In conventional opacity with a passive intruder, the observer is constructed to track the system states based on the observable events [15]. In the reactive opacity formalism, however, the intruder knows the injected input word, and hence the system (non-deterministic) transitions. As it is illustrated in Example 3 the active intruder can utilize the system observable responses to resolve the ambiguity of his estimation caused by the system’s non-deterministic transition. The observer for RCSO verification therefore, should include both possible input and observable output behavior of the system to track the estimated states. Furthermore, an open DES may only have a single and perhaps unique unobservable output event for a given input that can reveal a secret state. Therefore, in contrary to the conventional opacity with passive intruder, an active intruder can even use an unobservable response to infer the open DES states. This ability should be encoded in the active intruder observer.

**Definition 6 (Observer for RCSO):** Given an open DES \( G = (Q, X, \Delta, Q_0, T, \lambda) \), a projection function \( P \) with respect to the observable output events \( \Delta_o \), the observer automata is a deterministic finite-state automata \( G_o = Ac(Q, X, \Delta_o, Q_0, T_o) \) with state set \( Q = 2Q \), the initial state set is \( Q_0 = Q_0 \cup \{ q \in Q \mid 3q_0 \in Q_0 \} \), s.t. \( q \in T(q_0, \epsilon) \).

Let’s denote \( \Delta_{o, \epsilon} = \Delta_o \cup \{ \epsilon \} \), the transition function is \( T_o: Q \times X \times \Delta_{o, \epsilon} \rightarrow Q, \) that for any \( q \in Q, x \in X, \) and an observable event \( \delta \in \Delta_o \) is given by \( T_o(q, x, \delta) = \{ q' \in Q \mid 3q \in q' \in T(q, x) \text{ and } \delta \in \lambda(x, q') \} \), for an unobservable event, it is defined by \( T_o(q, x, \epsilon) = \{ q' \in Q \mid 3q \in \lambda(x, q') \} \) s.t. \( \delta_{o, \epsilon} \in \Delta_{o, \epsilon} \).

The initial estimated states \( Q_0 \) is constructed based on the combination of the possible initial states, \( Q_0 \), and any initial transitions with no input to the open DES, i.e., \( T(q_0, \epsilon) \). Note that, based on the definition of open DES in Definition 1 for any \( q_0 \in Q_0 \), we have \( \epsilon \in \lambda(q_0, \epsilon) \), and therefore, \( Q_0 \) is
solely defined based on $Q_0$ and $T(q_0, \epsilon)$. In the constructed observer, $T_o(q, (x, \epsilon))$ captures the active intruder ability to infer the system transition when he injects input $x$ and receives no observable output.

Given the constructed observer $G_o$, one can verify if $G$ is RCS-opaque by checking if there exists any state $\tilde{q} \in \tilde{Q}$ which is reachable from $Q_0$ and only contains the system secret states $Q_s$, i.e., $\tilde{q} \in Q_s$. The RCSO verification based on the proposed observer construction is formally given in the following theorem.

**Theorem 1:** Given an open DES $G = (Q, X, \Delta, Q_0, T, \lambda)$, the projection function $P$, the secret state set $Q_s \subseteq Q$, the associated observer $G_o = A\epsilon(Q, X, \Delta, Q_0, T_o)$ can be constructed by following Definition 6. Then $G$ is RCS-opaque if and only if for all $\tilde{q} \in \tilde{Q}$ either $\tilde{q} = \emptyset$ or $\tilde{q} \notin Q_s$ holds.

**Proof:** Necessary: here we show if $G$ is RCS-opaque, then there is no state $\tilde{q} \in \tilde{Q}$ in the constructed observer (following Definition 6) that $\tilde{q} \notin Q_s$. Let’s denote $Q_o \subseteq \tilde{Q}$ as the reachable states in $G_o$. To prove this part, we only need to show that for any input word and the observed output word, the states in the observer $G_o$ are the estimated current-state of the system. Consider any $\rho \in (X \times \Delta)^*$, such that $T_o(Q_0, \rho)$, then since $\rho \in P_{X\Delta}(L_{io}(G))$, there should exists $w \in L(G)$ and $\alpha \in P(O(w))$ such that $P_x(\rho) = w$, $P_{\Delta}(\rho) = \alpha$, and $\tilde{Q}(w, \alpha) \neq \emptyset$. In addition, following Definition 6 $\tilde{Q}(w, \alpha)$ and $T_o(Q_0, \rho)$ provides the same estimated states, meaning, for any $q \in Q(w, \alpha)$, we have $\tilde{q} = T_o(Q_0, \rho)$ with $q \in q$. Therefore, if $G$ is RCS-opaque, then $\tilde{Q}(w, \alpha) \notin Q_s$, which implies $\tilde{q} \notin Q_s$.

Sufficiency: here we show if for all $\tilde{q} \in Q_o$, we have $\tilde{q} \notin Q_s$ then $G$ should be RCS-opaque. We prove this part by contradiction. Let’s assume $G$ is not RCS-opaque that implies there should exists a $w \in L(G)$ such that $\tilde{Q}(w, \alpha) \subseteq Q_s$ for some $\alpha \in P(O(w))$. Therefore, similar to the necessary part, we know $\tilde{Q}(w, \alpha) = T_o(Q_0, \rho)$ with $P_x(\rho) = w$ and $P_{\Delta}(\rho) = \alpha$, provide the same estimated states. This implies, we have the observer state $\tilde{q} = T_o(Q_0, \rho)$ that $\tilde{q} \subseteq Q_s$, which contradicts the first assumption.

The following example illustrates the observer construction described above.

**Example 4:** Consider the open DES $G$ in Figure 4 with $Q_s = \{3\}$, $\Delta_o = \{\delta_1, \delta_2, \alpha\}$, and $\Delta_{\text{obs}} = \{b\}$. The constructed observer for $G$ is shown in Figure 4. An edge label is in the form of $x, \delta$, where $x \in X$, and $\delta \in \Delta_{\text{obs}}$. As it is shown in the Figure 4 the secret state $\{3\}$ is reachable from the initial state in the constructed observer, indicating that $G$ is not RCS-opaque.

VI. CONCLUSION

In the conventional opacity formalism, the intruder is considered as a passive observer. In this paper, we studied opacity in the presence of an active intruder which beyond a passive observation, is capable of manipulating the system behavior. In this setup, the active intruder can inject a certain input to the system and combine it with the observed system response to infer the secrets. We therefore introduced reactive opacity notions which characterize a property that regardless of how the intruder selects the input word, the system’s secret property remains indistinguishable from the non-secrets. We furthermore showed that all the proposed reactive opacity notions can be transformed into the RCSO. Given a RCSO notion and a system modeled as NFT, we proposed an automata-based method to verify if the system respects RCSO requirements. In the future works, we plan to study probabilistic reactive opacity for stochastic DESs.

**REFERENCES**

[1] R. Focardi and R. Gorrieri, “A taxonomy of trace-based security properties for ccs,” in Computer Security Foundations Workshop VII, 1994. CSFW 7. Proceedings. IEEE, 1994, pp. 126–136.

[2] F. Liu, “Opacity of discrete event systems and its applications,” Automatica, vol. 47, no. 3, pp. 496–503, 2011.

[3] R. Jacob, J.-J. Lesage, and J.-M. Faure, “Overview of discrete event systems opacity: Models, validation, and quantification,” Annual reviews in control, vol. 41, pp. 135–146, 2016.

[4] A. Saboori and C. N. Hadjicostis, “Notions of security and opacity in discrete event systems,” in Decision and Control, 2007 46th IEEE Conference on. IEEE, 2007, pp. 5056–5061.

[5] ——, “Verification of initial-state opacity in security applications of discrete event systems,” Information Sciences, vol. 246, pp. 115–132, 2013.

[6] X. Yin and S. Laforetune, “A new approach for the verification of infinite-step and k-step opacity using two-way observers,” Automatica, vol. 80, pp. 162 – 171, 2017.

[7] A. Bohannon, B. C. Pierce, V. Sjöberg, S. Weirich, and S. Zdancewic, “Reactive noninterference,” in Proceedings of the 16th ACM conference on Computer and communications security, 2009, pp. 79–90.

[8] A. Partovi and H. Lin, “Reactive supervisory control of open discrete event systems,” in 2009 IEEE 58th Conference on Decision and Control (CDC). IEEE, 2009, pp. 1056–1061.

[9] K. R. O’Neill, M. R. Clarkson, and S. Chong, “Information-flow security for interactive programs,” in 19th IEEE Computer Security Foundations Workshop (CSFW’06). IEEE, 2006, pp. 12–pp.

[10] J. W. Bryans, M. Koumy, L. Mazaré, and P. Y. Ryan, “Opacity generalised to transition systems,” in International Workshop on Formal Aspects in Security and Trust. Springer, 2005, pp. 81–95.

[11] M. Mohri, “Weighted finite-state transducer algorithms. an overview,” in Formal Languages and Applications. Springer, 2004, pp. 551–563.

[12] A. Khalili and A. Tacchella, “Learning nondeterministic mealy machines,” in International Conference on Grammatical Inference, 2014, pp. 109–123.

[13] A. Bouajjani, B. Jonsson, M. Nilsson, and T. Touili, “Regular model checking,” in International Conference on Computer Aided Verification. Springer, 2000, pp. 403–418.

[14] C. G. Cassandras and S. Laforetune, Introduction to discrete event systems. Springer Science & Business Media, 2009.

[15] C. N. Hadjicostis and C. Keroglou, “Opacity formulations and verification in discrete event systems,” in Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA). IEEE, 2014, pp. 1–12.

[16] Y.-C. Wu and S. Laforetune, “Comparative analysis of related notions of opacity in centralized and coordinated architectures,” Discrete Event Dynamic Systems, vol. 23, no. 3, pp. 307–339, 2013.