March 2008 UILU-ENG-08-2204
DC-234

DELAYED STATE ESTIMATION IN DISCRETE EVENT SYSTEMS AND APPLICATIONS TO SECURITY PROBLEMS

Anooshiravan Saboori and Christoforos N. Hadjicostis

Coordinated Science Laboratory
1308 West Main Street, Urbana, IL 61801
University of Illinois at Urbana-Champaign

REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
March 2008

3. REPORT TYPE AND DATES COVERED

4. TITLE AND SUBTITLE
Delayed State Estimation in Discrete Event Systems and Applications to Security Problems

5. FUNDING NUMBERS
NSF ECS 04-26831

6. AUTHOR(S)
Anooshiravan Saboori and Christoforos N. Hadjicostis

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
1308 West Main Street
Urbana, Illinois 61801-2307

8. PERFORMING RGANIZATION REPORT NUMBER
UILU-ENG-08-2204
DC-234

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Science Foundation
4201 Wilson Blvd
Arlington, VA 22203

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official position, policy, or decision, unless so designated by other documentation

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Application of discrete event systems in modeling and analyzing security problems has given rise to applications that require keeping track of (part of the) sequence of states that have been visited so far. Specifically, the notion of opacity requires that the truth of a certain predicate on the system state cannot be determined by an outside observer for the duration of a certain time window (or even at all times). Depending on the notion of opacity that is used, this predicate can be defined for states visited in the past (with no bound on how far into the past) or for states which have been visited a fixed number of observations in the past. In this report, motivated by such questions we introduce the problem of delayed estimation in discrete event systems modeled as a finite automaton with a finite number of states, unknown initial state, and partial event observation (but no state observation). Specifically, we consider two estimation problems: (i) initial state estimation which requires the estimate of the initial state following a sequence of observations and, (ii) K- delayed state estimation which requires the estimate of the state the system was in when it generated the Kth to last output (i.e., the state of the system K observations ago). To solve these two problems we construct appropriate state estimators and show that these delay state estimators can be used to verify opacity notions of interest.

14. SUBJECT TERMS
Discrete Event Systems, State Estimation, Automata, Security

15. NUMBER OF PAGES
45

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT
UL

Delayed State Estimation in Discrete Event Systems and Applications to Security Problems*

Anooshiravan Saboori and Christoforos N. Hadjicostis†

March 4, 2008

*This material is based upon work supported in part by the National Science Foundation under NSF Career Award No 0092696 and NSF ITR Award No 0426831. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of NSF.

†The authors are with the Coordinated Science Laboratory, and the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, IL 61801–2307, USA. Corresponding author: C. N. Hadjicostis, 357 CSL, 1308 West Main Street, Urbana, IL 61801–2307, USA (e-mail: `chadjic@uiuc.edu`).

1 Introduction

The problem of state estimation in *discrete event systems* (DES) along with its applications to fault diagnosis and control, has attracted significant research interest over the last two decades [1]–[8]. This problem consists of reconstructing all possible states that a known system can be in based on possibly limited knowledge of its initial state and partial knowledge of the sequence of events that occur in the system. The state estimation problem has found applications in diverse areas, including stabilizing supervisory control [7],[9], fault diagnosis [3],[8], interface design [5], and discrete event system inversion [10]. In what follows, we briefly describe the role that state estimation plays in these applications.

In the state-based supervisory control problem, the control objective requires that the sequence of states visited satisfies one of the following: (i) it follows a certain trajectory (tracking problem) [11], or (ii) it visits certain states frequently often (stabilizability problem) [9], or (iii) it avoids visiting certain states (state avoidance problem) [12]. In all cases, the control objective is achieved by enabling/disabling controllable events at appropriate times; depending on the control objective, these problems transforms to different observability conditions that commonly require that the *current* state can be uniquely determined at all times [7] or perhaps at times separated by a bounded number of transitions [13]. In fault diagnosis applications, diagnosability requires that the evolution of states to the faulty states be detected with a bounded delay. It is assumed that once the system reaches those faulty states, it never leaves them. Diagnosability is verified using a *diagnoser* which is a discrete structure that essentially acts as an estimator of the possible *current* states of the given system, making sure that the fault status of the estimated states is propagated appropriately [3][8]. In interface design, the goal is to design the interface so as to provide enough information to represent a true abstraction of the system without overwhelming a human operator with unnecessary information. This requirement is translated to *immediate observability* which requires that the current state of the system can be uniquely determined from the output associated with the current state and the last or next event [5]. Finally, a discrete event system is invertible with delay if it is possible to reconstruct (with some fixed delay K) the full event sequence given the observation sequence [10]. The state estimate is used (along with the plant model) to reconstruct the executed sequence of events. Also note that, the notion of observability with delay and the notion of a state estimator with delay which are introduced in [10] are used along with invertibility with delay to characterize a sufficient condition for *resilient invertibility with delay*, a notion that requires invertibility with delay even when observations may be corrupted.

In order to achieve the desired goal in all of the above applications, an estimate of the *current* state is sufficient and thus past information about the sequence of states visited so far is discarded. Therefore, the state estimator structure used, resembles the structure in [6] which captures the estimate of the current state of the system (in terms of our notation here the structure in [6] is a zero-delay state estimator). More recently, the use of discrete event system in modeling and analyzing security problems has given rise to applications that require keeping track of (part of the) sequence of states that have been visited so far [14],[15]. Specifically, the notion of *opacity* requires that the truth of a certain predicate on the system state cannot be determined by an outside observer for the duration of a certain time window (or even at all times). Depending on the notion of opacity that is used, this predicate can be defined for states visited in the past (with no bound on how far into the past) or for states which have been visited a fixed number of observations in the past. In either case, existing state estimation techniques cannot verify these properties since they are tracking the current state but not the trajectory of the states. In this report, motivated by questions that arise frequently in security and privacy applications, we introduce the problem of *delayed estimation* and construct delay state estimators which are capable of capturing additional information about state estimates and can be used to verify opacity notions of interest.

The basic state estimation setting we consider is the following: We are given a finite automaton with a finite number of states, unknown initial state, and partial *event* observation (but no *state* observation). For this type of DES, we consider two estimation problems: initial state estimation and K-delayed estimation. The former requires the estimate of the initial state following a sequence of observations whereas the latter requires the estimate of the state the system was in when it generated the K^{th} to last output (i.e., the state of the system K observations ago). To solve these two problems we construct appropriate state estimators. Specifically, we construct (i) an *initial-state estimator* (that can be used to capture all the information that is relevant to initial state estimation and is contained in any sequence of observations of finite but arbitrary length); (ii) a K-

delay state estimator (that can be shown to contain, after observing any sequence of observations, the information that is necessary to deduce the state the system was in K observations ago). Note that the K-delay state estimator and the initial-state estimator are not comparable since they aim to capture different information.

Our work is related to the inverter with delay that was introduced in [10]. Assuming that the system is invertible with delay, the inverter in [10] acts as an *online* algorithm which, at any time, stores the K future observations (where K is the fixed delay in the definition of the invertibility with delay) in order to later be able to distinguish the states at the current time (using back propagation). The refined state estimate that is obtained is used along with the plant model to reconstruct the executed sequence of events. In this report, we propose a finite state structure that can *capture* estimates with delay for a future observation sequence of any length. In other words, what we do here can be seen as an *offline* approach for refining the current state estimate using all possible sequences of K future observations.

This report is organized as follows: Section 2 covers the preliminaries and background. In Section 3 we define three notions of opacity and provide a motivational example for delayed state estimation. Section 4 formally defines delayed estimation and initial-state estimation problems, and introduces the K-delay state estimator and the initial-state estimator as respective solutions to these problems; this section also studies the structural relation between these two state estimators. In Section 5, we provide verification methods for different notions of opacity using state estimators; we also derive sufficient conditions under which these definitions become equivalent. Section 6 concludes the report and provides some directions for future research.

2 Preliminaries and Notation

Let Σ be an alphabet of symbols (also called elements or events) and denote by Σ^* the set of all finite-length strings of elements of Σ, including the empty string ϵ (of length zero). A language $L \subseteq \Sigma^*$ is a subset of finite-length strings from Σ^*. For a string ω, $\overline{\omega}$ denotes the *prefix-closure* of ω and is defined as $\bar{\omega} = \{t \in \Sigma^* | \ \exists s \in \Sigma^* : ts = \omega\}$. The post-string ω/s of ω after s is defined as $\omega/s = \{t \in \Sigma^* | \ st = \omega\}$. Similarly, the pre-string of ω before s is defined as $\omega \backslash s = \{t \in \Sigma^* | \ ts = \omega\}$.

For any string t, $|t|$ denotes the length of t [16, 17].

A DES is modeled in this report as a finite automaton $G = (X, \Sigma, \delta)$, where $X = \{0, 1, \dots, N-1\}$ is the set of states, Σ is the set of events, and $\delta : X \times \Sigma \to X$ is the (possibly partial) state transition function.[1] Note that in our model the initial state of the DES G is not known and is taken to be X. The function δ can be extended from the domain $X \times \Sigma$ to the domain $X \times \Sigma^*$ in a (routine) recursive manner

$$\delta(i, \epsilon) := i,\ i \in X$$
$$\delta(i, ts) := \delta(\delta(i, t), s) \text{ for } s \in \Sigma^* \text{ and } t \in \Sigma$$

(Note that $\delta(i, s)$ for $s \in \Sigma^*$ is undefined if any of the transitions in the recursion is undefined.) The behavior of DES G is captured by

$$L(G) := \{s \in \Sigma^* \mid \exists i \in X, \delta(i, s) \text{ is defined}\}.$$

We use $L(G, i)$ to denote the set of all traces that originate from state i of G (so that $L(G) = \bigcup_{i=0}^{N-1} L(G, i)$).

We assume that only a subset Σ_{obs} of the events in Σ can be observed and adopt the common assumption that Σ can be partitioned into two sets, Σ_{obs} and Σ_{uo}. The natural projection $P_{\Sigma_{obs}} : \Sigma^* \to \Sigma_{obs}^*$ can be used to map any trace executed in the system to the sequence of observations (observable transitions) associated with it. This projection is defined recursively as $P_{\Sigma_{obs}}(\sigma s) = P_{\Sigma_{obs}}(\sigma) P_{\Sigma_{obs}}(s),\ \sigma \in \Sigma, s \in \Sigma^*$, with

$$P_{\Sigma_{obs}}(\sigma) = \begin{cases} \sigma & \text{if} \quad \sigma \in \Sigma_{obs}, \\ \epsilon & \text{if} \quad \sigma \in \Sigma_{uo} \cup \{\epsilon\}, \end{cases}$$

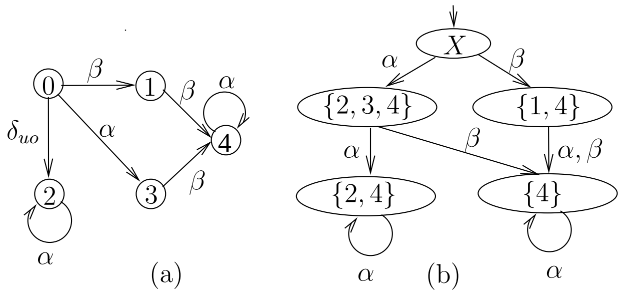
where ϵ represents the empty string [16, 17]. In the sequel, the index Σ_{obs} in $P_{\Sigma_{obs}}$ will be dropped when it is clear from context.

[1] If the transition function is defined as $\delta : X \times \Sigma \to 2^X$ (where 2^X is the power set of X), then the DES is *nondeterministic*.

Upon observing some string (sequence of observations) s, the state of the system might not be identifiable uniquely due to the lack of knowledge of the initial state and the partial event observation. We denote the set of states that the system might reside in *given that s was observed* as the (current) state estimate. The *zero-delay state estimator* is an automaton $G_{0,obs}$ which captures these estimates and can be constructed as follows [6]: each state of this automaton 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 X_0 of $G_{0,obs}$ is X taken to be representing the fact that nothing is assumed about the initial state (i.e., the initial state could be any state). At any state Z of the state estimator ($Z \subseteq X$), the next state upon observing an event $\sigma \in \Sigma_{obs}$ is the unique state of $G_{0,obs}$ associated with the set of states that can be reached from the states in Z with a string of events that generates the observation s. We denote this automaton by $G_{0,obs} = AC(2^X, \Sigma_{obs}, \delta_{obs}, X_0)$ where 2^X (power set of X) is the state set, Σ_{obs} is the set of observable events, δ_{obs} is the transition operator, X_0 denotes the initial state (taken to be $X_0 \equiv X$) and, AC denotes the part of the automaton that is accessible from initial state X_0. We also define $X_{obs} \subseteq 2^X$ to be the reachable states from X_0 under δ_{obs} so that $AC(2^X, \Sigma_{obs}, \delta_{obs}, X_0) = (X_{obs}, \Sigma_{obs}, \delta_{obs}, X_0)$. The following example clarifies this construction.

Example 1. *Consider the DES G in Figure 1-a. Assuming that $\Sigma_{obs} = \{\alpha, \beta\}$, then the zero-delay state estimator $G_{0,obs}$ is constructed as follows. The initial state is $X_0 = X$. Next, starting from the initial state X_0 and observing α, the current state is any of the states in $\{2, 3, 4\}$; at this new state, the set of possible transitions is the union of all possible transitions for each of the states in $\{2, 3, 4\}$. Note that if instead of α, we observe β from the initial state X_0, the set of possible states is $\{1, 4\}$. Following this procedure, $G_{0,obs}$ can be constructed as in Figure 1-b. Clearly, X_{obs} in this case is $\{\{4\}, \{1, 4\}, \{2, 4\}, \{2, 3, 4\}, \{0, 1, 2, 3, 4\}\}$.*

Any $m \in 2^{X^2}$ is a subset of 2^{X^2} and contains some pairs of states. In this report, m will be viewed as a *state mapping* consisting of a starting state and an ending state. The set of states included as the first (second) component in these pairs is called the set of starting (ending) states of m. We denote the set of starting states for state mapping m by $m(1)$ and the set of ending states by $m(0)$. Since we use the notion of state mapping frequently in this report, we provide some



(a) (b)

Figure 1: (a) G; (b) $G_{0,obs}$.

definitions related to it. We say that state mapping m_1 *refines* m_2 if the set of starting states of m_1 is a subset of the set of starting states of m_2. Moreover, if mapping m_1 refines m_2 and mapping m_2 refines m_1, then we say that mapping m_1 is *consistent* with m_2. We also 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

$$m_1 \circ m_2 := \{(i_1, i_3) | \exists i_2 \in X, (i_1, i_2) \in m_1, (i_2, i_3) \in m_2\}.$$

The composition operator takes as inputs two sets of 2-tuples and produces as output another set of 2-tuples by including all 2-tuples with the first element borrowed from a 2-tuple in the first input set and the second element borrowed from a 2-tuple in the second set, as long as these two 2-tuples share the same remaining (second/first) element. Note that this operator is only defined for tuples of size two.

We can map any observation of finite but arbitrary length in DES G to a state mapping by using the mapping $M : \Sigma_{obs}^* \rightarrow 2^{X^2}$ defined as

$$M(s) = \{(i, j) | i, j \in X, \exists t \in \Sigma^*, P(t) = s, \delta(i, t) = j\},$$

which we call the s-induced state mapping. If $(i, j) \in M(s)$, then there exists a sequence of events that starts from state i and ends in state j, and produces observation s. We also define the binary relation R on $\Sigma_{obs}^* \times \Sigma_{obs}^*$ as sRt if and only if $M(s) = M(t)$. Clearly R is an equivalence relation and, thus, induces a partition on Σ_{obs}^*. The number of equivalence classes induced by R is at most 2^{N^2} where N denotes the number of states. Finally, for any $Z \subseteq X$, we define the operator

$\odot : 2^X \to 2^{X^2}$ to represent $Z \odot Z := \{(i,i) | i \in Z\}$.

Given a finite automaton $G = (X, \Sigma, \delta)$, we define X^K ($K \geq 2$) as the set of K-tuples of states of DES G, i.e.

$$\begin{aligned} X^K &:= X \times X \times \ldots \times X \\ &= \{(i_1, \ldots, i_K) | i_k \in X, 1 \leq k \leq K\} \end{aligned}$$

and

$$X^* := \bigcup_{k=2}^{\infty} X^k.$$

A trellis is a graph whose nodes are ordered into vertical slices and each node at each time is connected to (at least) one node at an earlier time and (at least) one node at a later time. Any $m \in 2^{X^K}$ can be graphically represented by a trellis graph, hence we call m a trellis mapping of length K. Note that a state mapping is a special case of trellis mapping for $K = 2$. As in the case of state mappings, we can define the set of starting and ending states for a trellis mapping $m \in 2^{X^K}$. To keep the notation consistent with our earlier definition of state mappings, we use $m(K-1)$ to denote the set of starting states and $m(0)$ to denote the set of ending states. We also denote by $m(k), 1 < k < K-1$, the set of intermediate states in the K-tuple. The definitions of refinement and consistency can be carried on in a manner similar to the manner used in the case of state mappings. We define the *shift* operator $\| : 2^{X^*} \times 2^{X^2} \to 2^{X^*}$ for trellis mapping $m_1 \in 2^{X^*}$ and $m_2 \in 2^{X^2}$ as

$$m_1 \| m_2 := \{(i_2, \ldots, i_K, i_{K+1}) | \exists i_1 \in X, (i_1, i_2, \ldots, i_K) \in m_1, (i_K, i_{K+1}) \in m_2\}.$$

Note that the shift operator takes as input two sets of tuples, the first set involves K-tuples $K \geq 2$, and the second involves tuples of size two; for each K-tuple of the first set, all 2-tuples in the second set whose first element is the same as the last element of the K-tuple from the first set are used to produce the output K-tuple by using: the 2^{nd}, 3^{rd}, ..., K^{th} elements of the first K-tuple and the second element of the second 2-tuple to create a new K-tuple as an output.

Generalizing the notion of the s-induced state-mapping, we map an observation s in DES G to a trellis mapping via the mapping $T_K : \Sigma_{obs}^* \rightarrow 2^{X^{K+1}}$

$$T_K(s) = \{(i_0, i_1, \ldots, i_K) | (0 \leq k \leq K, 0 \leq l \leq K-1) : i_k \in X, t_l \in \Sigma^*, \\ P(t_0 t_1 \ldots t_{K-1}) = s, \delta(i_l, t_l) = i_{l+1}\}, \quad (1)$$

which we call the s-induced trellis mapping. Finally, for any $Z \subseteq X$ and $K \geq 2$, we define the operator $\odot_K : 2^X \rightarrow 2^{X^{(K+1)}}$ product to represent $Z \odot_K Z = \{(i, i, \ldots, i) | i \in Z\}$ where the tuples involve $K+1$ identical elements. The following example illustrates the concepts of state and trellis mappings.

Example 2. *In this example, we consider the DES G represented in Figure 1-a with $\Sigma_{obs} = \{\alpha, \beta\}$ and $X = \{0, 1, 2, 3, 4\}$. First we construct the α-induced state mapping, i.e., $M(\alpha)$. Observe that α can be observed only from states 0, 2 and 4; moreover, if the initial state was 0, the current state can be any of the states in $\{2, 3\}$ but if the initial state was 2, the current state could only be $\{2\}$; similarly, if the initial state was 4, the current state would be 4. Hence $M(\alpha) = \{(0, 2), (0, 3), (2, 2), (4, 4)\}$. Following the same reasoning as in the case of $M(\alpha)$, we have $M(\beta) = \{(0, 1), (1, 4), (3, 4)\}$. We can compose these two state mappings as*

$$M(\alpha) \circ M(\beta) = \{(0, 2), (0, 3), (2, 2), (4, 4)\} \circ \{(0, 1), (1, 4), (3, 4)\} \\ = \{(0, 4)\}. \quad (2)$$

Next we consider the notion of induced trellis mapping. Upon observing α, the (3-tuple) induced trellis mapping $T_2(\alpha)$ becomes $\{(0, 0, 2),\ (0, 0, 3), (2, 2, 2), (4, 4, 4)\}$ following the same approach as in the state mapping case. We can shift $T_2(\alpha)$ with the state mapping induced by β, i.e., $\{(0, 1), (1, 4), (3, 4)\}$ to obtain another 3-tuple as $T_2(\alpha) \| M(\beta) = \{(0, 3, 4)\}$. □

3 Motivation

The exchange of vital information over shared cyber-infrastructures in many application areas (ranging from defense and banking to health care and power distribution systems) has led to concerns about the vulnerability of such systems to intruders and other malicious entities. As a result, various notions of *security* have received considerable attention from researchers and work pursued so far can be roughly classified into two main categories: the first approach focuses on carefully characterizing the intruder's capabilities whereas the second one focuses on the *information flow* from the system to the intruder [18],[19]. *Opacity* is a security notion that falls in the second category and aims to determine 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 to outsiders [14]. More specifically, this requires that an intruder (modeled as an observer of the system's behavior) is never able to establish the truth of the predicate.

In this section, we define opacity with respect to predicates that are state-based. More specifically, assuming that the system under consideration can be modeled as a finite automaton with partial observation on its transitions, we define the secret behavior of the system as the evolution of the system's state to a set of secret states. The intruder's ability to observe system activity is modeled through a fixed (static) projection map and, as a result, (delayed) state estimation becomes a crucial aspect for verifying these type of security/privacy properties. We first introduce these different state-based notions of opacity in Section 3.1,3.2 and 3.3; and then discuss their practical relevance in an example in Section 3.4.

3.1 Initial-State Opacity

The first notion of opacity considered in this section is initial-state opacity. In certain applications, such as encryption, some vital initial iteration information (e.g., the key used for encryption) should be kept secret from an outside observer for the *whole* length of the observation. We call this property initial-state opacity and we define it formally as follows.

Definition 1 (Initial-State Opacity)**.** *Given a finite automaton $G = (X, \Sigma, \delta)$, a projection map P with respect to the set of observable events Σ_{obs}, and a set of secret states $S \subseteq X$, automaton G is initial-state opaque with respect to S and P (or (S, P, ∞) initial-state opaque), if*

$$\forall i \in S, \forall t \in L(G, i) : \exists j \in X - S, \exists s \in L(G, j), P(s) = P(t).$$

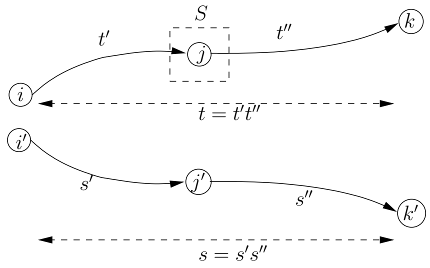


Figure 2: Graphical representation of the notation used in Definition 2 and in the proof of Theorem 4 ($|P(t'')| = |P(s'')| \leq K$).

According to Definition 1, the system G is (S, P, ∞) initial-state opaque if for every string t that originates from a state in the secret set S there exists a string s that originates from a state outside S and has the same projection as t. In other words, initial-state opacity requires that regardless of the string that might be generated by the system (and which will result to a corresponding sequence of observations) no information about the membership of the initial state of the system to the set of secret states S can be inferred.

3.2 K-Step Opacity

The next notion of opacity we consider is K-step opacity [15]. This notion is suitable for cases when, following a minimum of K observations, an outside observer is allowed to infer information about the set of secret states (because the secret transaction has completed or because the intrusion will be detected or because of other reasons). The following definition defines K-step opacity formally. Please refer to Figure 2 for a graphical presentation of the definition.

Definition 2 (K-Step Opacity)**.** *Given a finite automaton $G = (X, \Sigma, \delta)$, a projection map P with respect to the set of observable events Σ_{obs}, and a set of secret states $S \subseteq X$, DES G is K-step opaque (for some $K \geq 0$) with respect to S and P (or (S, P, K)-opaque), if for all $t \in \Sigma^*$, $t' \in \bar{t}$, and $i \in X$,*

$$\delta(i, t') \in S, \delta(i, t) \textit{ is defined}, |P(t)/P(t')| \leq K \Rightarrow \exists s \in \Sigma^*, \exists s' \in \bar{s}, \exists i' \in X, P(s) = P(t),$$

$$P(s') = P(t'), \delta(i', s') \notin S, \delta(i', s) \textit{ is defined.}$$

According to Definition 2, DES G is (S, P, K)-opaque if for every string t in $L(G)$ that passes through S within the past K observations, there exists a string s in $L(G)$ with $P(s) = P(t)$ such that when string t passes through S, string s does not. K-step opacity requires opacity for only K observations since the last entrance of the system to the set of secret states S.

3.3 Infinite-Step Opacity

Note that (S, P, K)-opacity requires opacity for only K observations since the last entrance of the system to the set of secret states. This notion is suitable for cases where there is a bounded delay, after which one does not care if the outside observer can infer information about the behavior that was considered previously secret. However, in many applications the existence of such bound might not be true. For this reason, we extend the definition of (S, P, K)-opacity to cases where $K \to \infty$. In other words, we require that the opacity of membership of states to the set of secret states remains valid for the whole length of the observation. Below, we first define infinite-step opacity formally.

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

$$\delta(i, t') \in S, \delta(i, t) \textit{ is defined} \Rightarrow \exists s \in \Sigma^*, \exists s' \in \bar{s}, \exists i' \in X,$$

$$P(s) = P(t), P(s') = P(t'), \delta(i', s') \notin S, \delta(i', s) \textit{ is defined.}$$

According to Definition 3, DES G is (S, P, ∞)-opaque if for every string t in $L(G)$ that passes through S, there exists a string s in $L(G)$ with $P(s) = P(t)$ such that when string t passes through S, string s does not. In other words, infinite-step opacity requires that no string that can potentially be generated by the system can provide explicit information regarding the entrance of the system to the set of secret states *at any point in time.*

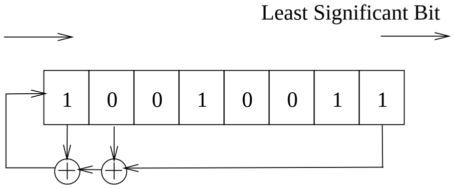


Figure 3: A conventional 8-bit LFSR with tapped bits 0,1,7 and seed (initial state) 10010011.

3.4 Motivational Example

The aforementioned state-based notions of opacity have been shown to be useful in modeling security properties in encryption, communication and secure protocols [14]. The following is a motivating example from the area of secure protocols for all of the above notions of opacity.

Example 3. *In cryptography, a symmetric cipher combines plain text (original information) bits with a pseudo-random bit stream (key-stream), typically using an XOR operation. For example, message* 1010 *XOR-ed with key-stream 0100 results in the encrypted message 1110. Knowledge of the encrypted message does not reveal the plain text unless the key-stream is compromised. To create the key-stream, one often uses a linear feedback shift register (LFSR) as a pseudo-random number generator (Figure 3). An LFSR is an autonomous shift register whose input (leftmost or most significant) bit is obtained by XOR-ing some predefined combination of the bits that are stored in the shift register. This implies that the input bit is a linear function of the LFSR's previous state. The initial state of the LFSR is called the seed, and the list of the bit positions that affect the next state is called the tap sequence. The taps are XOR-ed sequentially and then fed back into the register as the leftmost bit. Figure 3 shows an 8-bit LFSR with tapped bits 0,1,7, and seed 10010011. Because the operation of the register is deterministic, the sequence of values produced by the register (which is used as the key-stream for the stream cipher) is completely determined by its seed. For example, assuming that the seed (initial state) of the LFSR in Figure 3 is 10010011, then the next output is 1 (i.e., the rightmost bit shifted out) and the next state of the LFSR becomes 01001001 (because the incoming leftmost bit is given by* $1 \oplus 0 \oplus 1 = 0$ *and the rest of the bits are the leftmost seven bits of 10010011 with the rightmost bit shifted out). Note that the register has a finite number of possible states (*2^8 *states), so it must eventually enter a repeating cycle. An LFSR* *with a well-chosen feedback function (taps) and initial state*[2] *can have a very long cycle and can produce a sequence of bits which appears random. Alternative structures to the conventional LFSR do exist (see, for example, the idea of a clock mechanism in [20]).*

A5/1 is a stream cypher that is used to encrypt messages in GSM mobile phone systems. It is generated by a combination of three LFSRs with clocking mechanisms that was kept secret by GSM companies for a long time. Anderson [21] first identified and published the general structure of A5/1 and later, Briceno et. al. [22] reversed-engineered this protocol.

An intruder/observer can interact with this protocol by inserting some plain text and observing the ciphered text in order to find the seed. Note that finding the seed is equivalent to finding the stream of the keys that were used to encrypt all previous messages. Hence, if the intruder records all of the (encrypted) conversation, after finding the seed, he/she can go back and decrypt them using the key-stream. Clearly, many of the security concerns about this protocol can be recast in terms of the opacity notions considered in this report: is there a seed for which there exists a sequence of inputs that reveals that seed? We can obtain the answer to this question by formulating the problem as an initial-state opacity problem. Note that if there is such a seed, one might be interested in how long (in terms of input size) does it take for the intruder to detect it. An answer to this question can be obtained in terms of a problem formulation that involves K-step opacity (see [15]). Similarly, if there exists a sequence of inputs that reveals the state of the key-stream generator at some point in time (but not necessarily the initial state) then communication after this point in time can be compromised. An answer to this latter question can be obtained via the infinite-step opacity problem formulation. □

3.5 Use of Delay State Estimators in Verifying Opacity Properties

The two state estimators that we construct in Section 4 of this report can be used for verifying different notions of opacity. In Section 5, we show that K-step opacity (initial-state opacity) for a discrete event system can be verified using an appropriately constructed K-delay state estimator (initial-state estimator). For infinite-step opacity, we provide a verification method which involves a combination of initial state and zero-delay state estimators. We also study the relationship between

[2] Clearly, initial state 00000000 would not be a good choice in this example regardless of the choice for tapped bits.

these notions of opacity and show that infinite-step opacity (not surprisingly) implies initial-state opacity and K-step opacity. We also show (perhaps surprisingly) that there is a finite bound K^* such that, for $K > K^*$, K-step opacity and infinite-step opacity are equivalent. We show this by using our state estimator structure to prove that the information contained in the K-delay state estimator about states visited K observations into the past is the same as that of K'-delay state estimator for $K' > K \geq K^*$.

3.6 Related Work

There already exists some work on security in DES [23], [24], [25], and our work in this report is certainly related to it. In particular, the authors of [23] consider finite state Petri nets and define infinite-step opacity with respect to state-based predicates. Following a *language-based* approach, the authors of [24] consider multiple observers with different observation capabilities (modeled through different observable transitions); opacity in this setting requires that no observer is able to determine whether the actual trajectory of the system belongs to the secret language that is assigned to it. In both [23] and [24], for opacity to hold, the projection of secret trajectories needs to be verified to be a subset of the projection of the remaining trajectories in the system. The authors of [25] partition the event set into high level and low level events and consider the verification problem of *intransitive interference* which captures the allowed information flow (e.g., occurrence of certain events) from the high level events to the low level events through a downgrading process.

Our work in this report essentially extends the notion of opacity defined in [23] for Petri nets to automata and also introduces the related notion of K-step opacity. Note that, in contrast to [24], opacity in our framework assumes that the states of the system can be partitioned into *secret* and *non-secret* ones; this state-based formulation is what enables us to use a state estimator to verify opacity. Our model of the intruder's capability (in terms of observability power) is different from [25] which makes the two frameworks hard to compare.

4 Delayed Estimation

As mentioned earlier, the system under consideration is a finite automaton with (partial) event observation and unknown initial state. In *delayed estimation*, we are interested in estimating the state of the system with a fixed delay (measured in terms of observation steps). More specifically, after observing a sequence of n labels, we are interested at the state of the system K labels ago; this is equivalent to the state of the system after observing $n-K$ labels *refined* by the knowledge of the K labels that followed.

Definition 4. *Given a finite automaton $G=(X,\Sigma,\delta)$ and a projection map P with respect to the set of observable events Σ_{obs}, assume that the string $s=\alpha_0\ldots\alpha_n$ is observed. Define the (true) system's state trajectory as the sequences of states that the system visits while generating observation s. We denote this trajectory by*

$$x_0,\ldots,x_m,$$

where $m\geq n$ implies that there might be some state transitions due to unobservable events and hence the number of states visited might be more than the number of observations. In (zero-delay) state estimation, we are interested in obtaining all the possible system's states along with the observation of s. We denote the sequence of the state estimates by

$$\tilde{X}_0,\ldots,\tilde{X}_{|s|},$$

where $\tilde{X}_t$ is the set of states that system could be in after observing $\alpha_0\ldots\alpha_{t-1}$ with $\tilde{X}_0=X$.

In delayed estimation, we are interested in refining state estimates with post observations. Denote the sequence of delayed estimates of the system's initial state along the observation by

$$\hat{X}_0(\epsilon),\hat{X}_0(\alpha_0),\hat{X}_0(\alpha_0\alpha_1),\ldots,\hat{X}_0(s)$$

with $\hat{X}_0(\epsilon)=\tilde{X}_0=X$. Here, $\hat{X}_0(\alpha_0\ldots\alpha_{t-1})$ is the set of states in $\tilde{X}_0$ from which $\alpha_0\ldots\alpha_{t-1}$ can *be observed. Similarly, we use*

$$\hat{X}_t(\alpha_0\alpha_1 \ldots \alpha_{t-1}), \hat{X}_t(\alpha_0\alpha_1 \ldots \alpha_{t-1}\alpha_t), \ldots, \hat{X}_t(s)$$

with $\hat{X}_t(\alpha_0\alpha_1 \ldots \alpha_{t-1}) = \tilde{X}_t$ *to denote the sequence of delayed estimates of the state of the system after observation* α_t*, starting as soon as* $\alpha_0\alpha_1 \ldots \alpha_{t-1}$ *is observed and continuing along the observation. In other words,* $\hat{X}_t(\alpha_0 \ldots \alpha_{t'-1})$ *for* $t' > t$ *is the set of states in* $\tilde{X}_t$ *from which* $\alpha_t \ldots \alpha_{t'-1}$ *can be observed. Finally, we can denote the sequence of delayed estimates of the system's state when* s *was observed by*

$$\hat{X}_{|s|}(s),$$

which satisfies $\hat{X}_{|s|}(s) = \tilde{X}_{|s|}$*. Using these definitions, we can denote the estimate of the system's state trajectory by*

$$\hat{X}_0(s) \ldots \hat{X}_{|s|}(s).$$

Note that the system's true state is always among the state estimates; also we necessarily have $\hat{X}_i(s) \subseteq \tilde{X}_i, i = 1, \ldots, |s|$, i.e., $\hat{X}_i(s)$ is essentially a refinement of $\tilde{X}_i$ that takes the post observations into consideration. Also, since $\tilde{X}_s$ does not vary with its post observations, we have omitted the argument s. The following example clarifies these definitions.

Example 4. *Consider DES* G *in Figure 4 with* $\Sigma_{obs} = \{\alpha, \beta, \theta\}$*. Here* G *is a finite automaton with initial state 0 and true state trajectory 0,1,0,2 (shown with a solid line in Figure 4-b) which generates the observation sequence* $\alpha\beta\theta$*. We assume that the observer does not have any prior information about the system true initial state (which was taken to be* 0*).*

Next we construct the sequence of the state estimates and the estimate of the system's state trajectory. Before any observation occurs, the state estimates and the estimate of the system's state trajectory is X*, i.e.,*

$$\hat{X}_0(\epsilon) = \tilde{X}_0 = X.$$

Upon the first observation, α*, this state estimate is updated as follows:*

$$\hat{X}_1(\alpha) = \tilde{X}_1 = \{0, 1, 2\}.$$

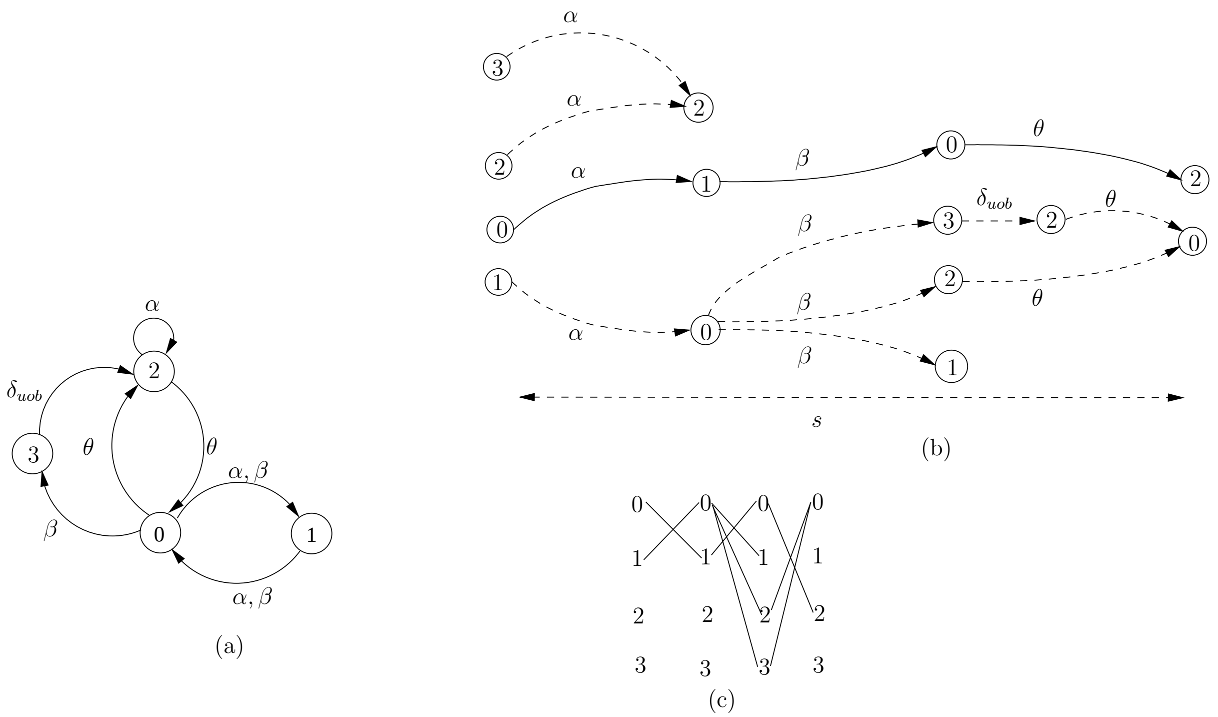


Figure 4: Constructions discussed in Example 4: (a) Automaton G modeling the system with $\Sigma_{obs} = \{\alpha, \beta, \theta\}$; (b) System's state evolution (solid line) and sequences of states with identical observation sequence (dashed lines); (c) Trellis graph corresponding to observation $\alpha\beta\theta$.

The explanation is straightforward: all states but state 3 in the system can be reached with this observation. Note that this observation does not change $\hat{X}_0(\epsilon)$, since α can originate from all of the states $\{0, 1, 2, 3\}$. Hence $\hat{X}_0(\alpha) = \hat{X}_0(\epsilon)$.

Next consider what happens after observing β. Following a similar discussion as above, one can justify the following

$$\hat{X}_2(\alpha\beta) = \tilde{X}_2 = X.$$

However, β can only be observed from states 0 and 1. Therefore, we need to update $\hat{X}_1(\alpha)$ according to this observation as

$$\hat{X}_1(\alpha\beta) = \{0, 1\}.$$

This update should be propagated back to update the initial state estimate as

$$\hat{X}_0(\alpha\beta) = \{0, 1\}.$$

Following a similar discussion, one can verify that upon observing $\alpha\beta\theta$, the estimate of the system's state trajectory $\hat{X}_0(\alpha\beta\theta), \ldots, \hat{X}_3(\alpha\beta\theta)$ is $\{0,1\}, \{0,1\}, \{0,2,3\}, \{0,2\}$, and the sequence of the state estimates $\tilde{X}_0, \ldots, \tilde{X}_3$ is $X, \{0,1,2\}, X, \{0,2\}$. A trellis graph that captures this information is shown in Figure 4-c. □

4.1 Initial State Estimation

Following this notation, we define the problem of initial state estimation as the problem of finding $\hat{X}_0(s)$ for any observation s.

Definition 5 (Initial-State Estimate). *Given a finite automaton $G = (X, \Sigma, \delta)$ and a projection map P with respect to the set of observable events Σ_{obs}, the initial state estimate after observing string s is defined as*

$$\hat{X}_0(s) = \{i | i \in X, \exists t \in \Sigma^*, P(t) = s, \delta(i,t) \textit{ is defined}\}.$$

The initial state estimation problem requires the initial state estimate after any sequence of observations (in other words, estimation of all states from which the sequence of observations could have been initiated). As Example 4 demonstrates, the first method that comes to mind is to use the sequence of observations to back-propagate the trajectory, and hence to estimate the initial state. Though straightforward, this method requires all of the observations to be sorted which is not feasible since the system might generate a sequence of arbitrary length (e.g., in an online application of this problem). Therefore, we need to find a way to map all possible sequences of observations to a set with finite elements so that we can store them with finite memory; clearly, the mapped elements need to contain the same amount of information regarding the initial state of the system as the original sequence of observations. Equivalently, we need to find a finite structure that allows us to process (perhaps online) information about the initial state as observations are coming in.

Next we introduce an algorithm which generates the initial-state estimator $G_{\infty,obs}$ using state mappings as its states. This state estimator starts from a state in which nothing about the system's initial state is known (namely, the state mapping associated with the initial state is $X_0 \odot X_0$ where

X_0 is the set of initial states of the system — taken to be $X_0 = X$ in our case). When the first observation is made, the induced state mapping corresponding to that observation is taken as the next state of the state estimator. After that, any observation causes $G_{\infty,obs}$ to transition to a state mapping which can be considered as the composition of the previous state mapping (associated with the state of $G_{\infty,obs}$) and the mapping induced by the new observation. The information captured by this composed state mapping (and thus by each state of $G_{\infty,obs}$) is the following: we know all pairs of one starting and one ending state such that we can reach the ending state from the starting state via a sequence of events that generated the observed sequence of events. This is all the information we need to keep as more labels are observed: at any given time, the state mapping gives all necessary information about the current state and the initial state estimates (and the connections between them) through its pairs of starting and ending states. Note that this structure is guaranteed to be finite and has at most 2^{N^2} states where N is the number of states of the discrete event system G.

The initial-state estimator, when considered as a finite state machine, summarizes the effect of the sequence of observations on the estimate of the initial state. This summary, along with a summary of the effect of the observations on the estimate of current state and possible paths between initial and current state estimates, is independent of the observation length; hence multiple observations might be mapped to the same state in the initial-state estimator, demonstrating the fact that they contain the same information regarding the initial-current state estimates and paths between them. This emphasizes the importance of state mapping, as a tool for compressing the information necessary and performing initial state estimation using finite memory. In other words, we can think of the state mappings as a way to partition the set of all observations, of arbitrary but finite length, into a finite number of equivalence classes. Two strings belong to the same equivalence class if they induce the same state mapping. The following algorithm describes this construction formally.

Definition 6 (Initial State Estimator (ISE))**.** *Given a finite automaton $G = (X, \Sigma, \delta)$ and a projection map P with respect to the set of observable events Σ_{obs}, we define the initial-state estimator as the deterministic automaton $G_{\infty,obs} = AC(2^{X \times X}, \Sigma_{obs}, \delta_{\infty,obs}, X_{\infty,0})$ with state set $2^{X \times X}$*

(power set of $X \times X$), event set Σ_{obs}, initial state $X_{\infty,0} = X \odot X$, and state transition function $\delta_{\infty,obs} : 2^{X \times X} \times \Sigma_{obs} \to 2^{X \times X}$ defined for $\alpha \in \Sigma_{obs}$ as

$$m' = \delta_{\infty,obs}(m, \alpha) := m \circ M(\alpha),$$

where $m, m' \in 2^{X \times X}$. Recall that $M(\alpha)$ denotes the state mapping that is induced by observing α at the beginning of the observation. Also, $\delta_{\infty,obs}$ can be extended to include strings in the usual manner. If we let $X_{\infty,obs} \subseteq 2^{X \times X}$ be the reachable states from the initial state $X_{\infty,0}$ under $\delta_{\infty,obs}$, then $G_{\infty,obs} = (X_{\infty,obs}, \Sigma_{obs}, \delta_{\infty,obs}, X_{\infty,0})$.

Note that $G_{\infty,obs}$ is a deterministic structure with initial state $X \odot X$. In the following Lemma, we show that the starting and ending states associated with a state of the state estimator $G_{\infty,obs}$ that is reached via a string s are respectively the set of states from which the observation s could have taken place and the set of states that can be reached from such initial states.

Theorem 1. *If state m in $G_{\infty,obs}$ (as constructed in Definition 6) is reachable from initial state $X_{\infty,0} = X \odot X$ via string s, then m is associated with a state mapping that satisfies*

$$m = \{(i, j) | i, j \in X, \exists t \in \Sigma^*, P(t) = s, \delta(i, t) = j\}.$$

Proof. Assume $s = \alpha_0 \ldots \alpha_n$ and denote the sequence of states visited in $G_{\infty,obs}$ via s by $m_0, \ldots, m_{n+1}$. We prove the result by induction: for $s = \alpha_0$, the statement is true by construction. Now assuming that the lemma holds for $s = \alpha_0\alpha_1 \ldots \alpha_{n-1}$, we prove it for $s = \alpha_0 \ldots \alpha_n$. Recall that m_{n+1} is a state in $G_{\infty,obs}$ reachable from the initial state with string s (in the Lemma state m_{n+1} is denoted by m). By construction, we have

$$\begin{aligned}
m_{n+1} &= m_n \circ M(\alpha_n) \\
&= \{(i,k)|\exists j \in X, (i,j) \in m_n, (j,k) \in M(\alpha_n)\} \quad \text{(definition of $\circ$ operator)} \\
&= \{(i,k)|\exists j \in X, i,k \in X, \exists t^n \in \Sigma^*, P(t^{n-1}) = \alpha_0\alpha_1\ldots\alpha_{n-1}, \delta(i,t^{n-1}) = j, \\
&\quad \exists (j,k) \in M(\alpha)\} \quad \text{(induction hypothesis)} \\
&= \{(i,k)|\exists j \in X, i,k \in X, \exists t^n \in \Sigma^*, P(t^{n-1}) = \alpha_0\alpha_1\ldots\alpha_{n-1}, \delta(i,t^{n-1}) = j, \\
&\quad \exists t_n \in L(G), P(t_n) = \alpha_n, \delta(j,t_n) = k\} \quad \text{(definition of $M(\alpha)$)} \\
&= \{(i,k)|i,k \in X, \exists t^n \in \Sigma^*, P(t^n) = \alpha_0\alpha_1\ldots\alpha_n, \delta(i,t^n) = k\}.
\end{aligned}$$

Note that in the last line, we use $t^n = t^{n-1}t_n$. If we rename k to j, t^n to t, and replace $\alpha_0\alpha_1\ldots\alpha_n$ with s; the proof is completed. □

Next we prove that the starting states in the state mapping associated with any state of $G_{\infty,obs}$ are the estimates of the initial state according to Definition 5.

Corollary 1. *The initial state estimate after observing s, $\hat{X}_0(s)$, can be captured using the ISE as follows: suppose $\delta_{\infty,obs}(X_{\infty,0}, s) = m$, then*

$$\hat{X}_0(s) = m(1).$$

Proof. By Theorem 1, $m = \delta_{\infty,obs}(X_{\infty,0}, s) = \{(i,j)|i,j \in X, \exists t \in \Sigma^*, P(t) = s, \delta(i,t) = j\}$. Therefore,

$$\begin{aligned}
m(1) &= \{i|(i,j) \in m\} \\
&= \{i|i \in X, \exists j \in X, \exists t \in \Sigma^*, P(t) = s, \delta(i,t) = j\} \\
&= \{i|i \in X, \exists t \in \Sigma^*, P(t) = s, \delta(i,t) \text{ is defined}\} \\
&= \hat{X}_0(s),
\end{aligned}$$

which completes the proof. □

The following example clarifies the ISE construction.

Example 5. *In this example, we consider the DES G represented in Figure 1-a with $\Sigma_{obs} = \{\alpha, \beta\}$. On the left of Figure 5, we show the initial-state estimator for this system. As mentioned earlier, the initial uncertainty about the initial state is assumed to be equal to the state space and hence $m_0 = X \odot X$. Upon observing α (and following the notation of Definition 6), the next state of the ISE becomes*

$$\begin{aligned} m' &= \delta_\infty(m_0, \alpha) \\ &= m_0 \circ M(\alpha) \\ &= \{(0,0),(1,1),(2,2),(3,3),(4,4)\} \circ \{(0,2),(0,3),(2,2),(4,4)\} \\ &= \{(0,2),(0,3),(2,2),(4,4)\} \\ &= M(\alpha) \\ &\equiv m_1. \end{aligned}$$

Example 2 explains how $M(\alpha)$ is synthesized. Note that on the right of Figure 5 we use a graphical way to describe the pairs associated with the ISE. Next, assume that we observe β which (following the same reasoning as in the case of $M(\alpha)$) results in $M(\beta) = \{(1,0),(1,1),(1,2),(3,1),(3,3)\}$. Based on the notation in Definition 6, we have

$$\begin{aligned} m' &= \delta_\infty(m_1, \beta) \\ &= m_1 \circ M(\beta) \\ &= \{(0,2),(0,3),(2,2),(4,4)\} \circ \{(0,1),(1,4),(3,4)\} \\ &= \{(0,4)\} \\ &\equiv m_4. \end{aligned}$$

Using this approach for all possible observations (from each state), the ISE construction could be completed as shown in Figure 5. □

Remark 1. *In the above discussions, it is assumed that the initial uncertainty about the initial*

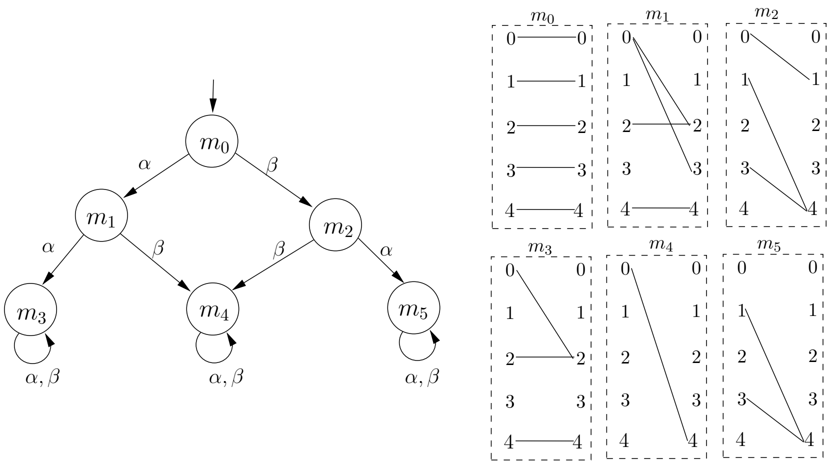


Figure 5: Initial-state estimator in Example 5.

state equals to the state space X (i.e., $\hat{X}_0(\epsilon) = X$). If this uncertainty can be reduced to a subset of the state space (i.e., $\hat{X}_0(\epsilon) = X_0 \subset X$), then we can easily modify the ISE construction to account for this extra information by changing the initial state to

$$X_{\infty,0} = X_0 \odot X_0.$$

Remark 2. *Note that all estimates of the initial state that are associated with a state (state mapping) m' that is reachable from a given state (state mapping) m in $G_{\infty,obs}$ are refinements of the estimates of the initial state associated with m. In other words, the initial state estimate, if changed, can only get more accurate; this is a straightforward conclusion of the definition of the initial state estimate and implies that observations that result in a traversal of a cycle in $G_{\infty,obs}$ do not yield extra information about the initial state. In other words, consider a cycle $m_1 \xrightarrow{\alpha_1} m_2 \xrightarrow{\alpha_2} \dots \xrightarrow{\alpha_n} m_n \xrightarrow{\alpha_{n+1}} m_1$ in $G_{\infty,obs}$ (constructed as in Algorithm 1). Also suppose that m_1 in the aforementioned cycle is reached via ω in $G_{\infty,obs}$, i.e., $\delta_{\infty,obs}(X_{\infty,0}, \omega) = m_1$. Then $m_1, m_2, \dots, m_n$ are consistent (refer to Section 2 for definition). Moreover observing ω, $\omega\alpha_1 \dots \alpha_{n+1}$ (or, in general, $\omega(\alpha_1 \dots \alpha_{n+1})^*$) yields the same information about initial state. This implies that one only needs to store s and to discard the other observations without loosing any information about the initial state. The use of state mappings is a way to systematically accomplish such reductions and hence provides us with a practical way to summarize observations (using finite storage).*

Remark 3. *In $G_{\infty,obs}$ there are cycles of states such that, once reached, there are no traces outside this cycle. We denote the states in these cycles as ergodic states. Ergodic states in ISE are associated with possible estimate about initial state that cannot be further refined. For example, the self-loop at state m_4 in Figure 5 denotes one such cycle (consisting of a single state). Note that if a sequence of observations reaches an ergodic state in ISE, the estimate of the initial state cannot be improved with future observations. For example, if we reach state m_4 via $\beta\beta(\alpha+\beta)^*$ (or $\alpha\beta(\alpha+\beta)^*$) there is no reason to wait for more observations. Note that in an automaton with n states, if we consider all strings of length n, at least one is guaranteed to reach an ergodic state. Also, since the ISE has at most 2^{N^2} states, we can argue that the set of all strings of length smaller or equal to $2^{N^2}-1$ reveal the same information about the initial state as any other set with length $K' > K$ (we elaborate on this when we discuss K-delayed estimation).*

4.2 K-Delayed Estimation

In the initial state estimation problem, we are interested in finding an estimate of the initial state which can be considered as the state estimate of a fixed point in the system's state trajectory. Another version of delayed estimation is to find an estimate of states of the system with a fixed delay with respect to the current observation. We denote this problem as K-delayed estimation [15]. Using our earlier notation, given observation $s = \alpha_0\alpha_1\ldots\alpha_n$, $n \geq k$, we are interested in finding $\hat{X}_{|s|-K}(s)$, which is called the K-delayed state estimate in [15]. The following definition describes this estimate formally.

Definition 7 (K-Delayed State Estimate)**.** *Given a finite automaton $G = (X, \Sigma, \delta)$ and a projection map P with respect to the set of observable events Σ_{obs}, the K-delayed state estimate after observing string $s = \alpha_0\alpha_1\ldots\alpha_n$ ($n \geq K$) is defined as*

$$\hat{X}_{|s|-K}(s) := \Big\{ i | i \in X, \exists s', s'' \in \Sigma^*, \exists j \in X : \delta(j, s') = i, \delta(i, s'') \text{ is defined, } P(s') = \alpha_0\alpha_1\ldots\alpha_{n-K},$$

$$P(s'') = \alpha_{n-K+1}\ldots\alpha_n \Big\}.$$

Based on Definition 7, the K-delayed state estimate $\hat{X}_{|s|-K}(s)$ after observing $s = \alpha_0\alpha_1\ldots\alpha_n$

$(n \geq K)$ is the set of all states that are reachable in G via a string s' which has a projection equal to the first $n-K$ observable events in s (in the same order) and for which there exists at least one continuation s'' with projection equal to the last K observable events in s (in the same order). Note that the set of states reachable in G via a string s' with projection $\alpha_0\alpha_1\ldots\alpha_{n-K}$ is the state estimate that was obtained after observing $\alpha_0\alpha_1\ldots\alpha_{n-K}$ but before observing $\alpha_{n-K+1}\ldots\alpha_n$ (i.e., the state estimate presented by a zero-delay (standard) state estimator). Using our earlier notation, this estimate is denoted by $\tilde{X}_{|s|-K}$. A K-delayed state estimate can be considered as the subset of these states from which the post K observations $\alpha_{n-K+1}\ldots\alpha_n$ are possible. Note that Definition 7 is the same as the definition of the "trajectory of state estimates" in [1]. The authors of [1] argue that the definition of always observability in [6] has shortcomings and use the "trajectory of state estimates" to define the notion of *always observability*. In [1], the construction of the trajectory of the state estimates is not considered.

We now discuss the construction of a finite automaton which is capable of constructing the K-delayed state estimates. Each state of this structure stores the trellis diagram induced by the last K observations (hence states can be considered as trellis mappings). Upon observing a new event, the current trellis mapping is shifted using the state mapping induced by the new observation. This procedure generates in general a new trellis mapping and hence a new state of the K-delay state estimator. Following this method the K-delay state estimator can be synthesized as a finite state machine. The finiteness of the state machine results from the finiteness of the trellis mappings that summarize the effect of the last K observations on the estimation of the sequence of states that have been visited. This summary is independent of the observation length and hence multiple observations might be mapped to the same trellis mapping in the K-delay state estimator, capturing the fact that they contain the same information about the K-delayed state estimate. As in the case of state mappings, we can think of a trellis mapping as a way to partition the set of all observations, of arbitrary but finite length, into a finite number of equivalence classes. Two strings belong to the same equivalence class if they induce the same trellis mapping. In this way, we can compress the information contained in the set of observations (which has infinitely many elements) using a finite set of trellis mapping without loosing the information contained in these strings about the K-delayed state estimate. In the following definition, we describe the construction of the K-

delay state estimator as the finite state machine which is capable of capturing the K-delayed state estimates.

Definition 8 (K-Delay Estimator)**.** *Given a finite automaton $G = (X, \Sigma, \delta)$ and a projection map P with respect to the set of observable events Σ_{obs}, we define the K-delay state estimator as the deterministic automaton $G_{K,obs} = AC(2^{X^{(K+1)}}, \Sigma_{obs}, \delta_{K,obs}, X_{K,0})$ with state set $2^{X^{(K+1)}}$, event set Σ_{obs}, initial state $X_{K,0} = X \odot_K X$, and state transition function $\delta_{K,obs} : 2^{X^{(K+1)}} \times \Sigma_{obs} \rightarrow 2^{X^{(K+1)}}$ defined for $\alpha \in \Sigma_{obs}$ as*

$$m' = \delta_{K,obs}(m, \alpha) := m \| M(\alpha),$$

where $m, m' \in 2^{X^{(K+1)}}$. Recall that $M(\alpha)$ denotes the state mapping that is induced by observing α at the beginning of the observation. Also, $\delta_{K,obs}$ can be extended to include strings in the usual manner. If we let $X_{K,obs} \subseteq 2^{X^{(K+1)}}$ be the reachable states from the initial state $X_{K,0}$ under $\delta_{K,obs}$, then $G_{K,obs} = (X_{K,obs}, \Sigma_{obs}, \delta_{K,obs}, X_{K,0})$.

Note that $G_{K,obs}$ is a deterministic structure with initial state $X \odot_K X$ which is essentially the set of $(K+1)$-tuples of the form $(i, i, \ldots, i)$ for $i \in X$. This corresponds to the fact that no information is available about the initial state. Also, each state of this automaton is essentially a trellis mapping of fixed size $K + 1$.

In the following Lemma, a state of $G_{K,obs}$ that is reached via a sequence of observations is associated with a set of $(K+1)$-tuples of states (i.e., a trellis mapping) such that the first $|s| - K$ observations would have taken the system to the starting states of the trellis mapping and in addition, the last K observations could have taken place from these starting states and visit the intermediate states in the tuple (in the order captured by the trellis mapping).

Lemma 1. *Suppose state m in $G_{K,obs}$ (as constructed in Definition 8) is reachable from the initial state $X_{K,0} = X \odot_K X$ via the string $s = \alpha_0 \alpha_1 \ldots \alpha_n$. Then, state m can be characterized as follows:*
(i) $|s| - K < 0$: $m = \{(i_0, i_1, \ldots, i_K) | (0 \leq k \leq K, 0 \leq l \leq |s| - 1, 0 \leq w \leq K - |s| - 2) : i_k \in X, i_w = i_{w+1}, \exists t_l \in \Sigma^*, P(t_l) = \alpha_l, \delta(i_{K-|s|+l}, t_l) = i_{K-|s|+l+1}\}$.

(ii) $|s| - K \geq 0$: $m = \{(i_0, i_1, \ldots, i_K) | (0 \leq k \leq K, 0 \leq l \leq K - 1) : i_k \in X, \exists t_l \in \Sigma^*, P(t_l) = \alpha_{n-K+1+l}, \delta(i_l, t_l) = i_{l+1}, \exists j \in X, t \in \Sigma^*, \delta(j, t) = i_0, P(t) = \alpha_0 \alpha_1 \ldots \alpha_{n-K}\}$.

Proof. (i) When $|s| < K$ (before K observations are made), the states in the s-induced trellis mapping (consisting of ($|s|+1$)-tuples) replace the rightmost states in the current $(K+1)$-tuple and hence the first $K-|s|$ leftmost states remain intact (and equal). This implies that

$$\begin{aligned}
m_n &= m_{n-1} \| M(\alpha_n) \\
&= \{(i_0, i_1, \ldots, i_K) | (0 \leq k \leq K, 0 \leq w \leq K-|s|-2) : i_k \in X, i_w = i_{w+1}, \\
&\quad \left(i_{K-|s|}, i_{K-|s|+1}, \ldots, i_K\right) \in M(\alpha_n)\} \\
&= \{(i_0, i_1, \ldots, i_K) | (0 \leq k \leq K, 0 \leq l \leq |s|-1, 0 \leq w \leq K-|s|-2) : i_k \in X, i_w = i_{w+1}, \exists t_l \in \Sigma^*, \\
&\quad P(t_l) = \alpha_l, \delta(i_{K-|s|+l}, t_l) = i_{K-|s|+l+1}\} \quad \text{(by (1))}.
\end{aligned}$$

This completes the proof of part (i).

(ii) Denote the sequence of states visited in $G_{K,obs}$ via s by $m_0, \ldots, m_n$. We prove the result by induction: for $s = \alpha_0 \alpha_1 \ldots \alpha_{K-1}$, ($|s| = K$), the statement is true by construction. Now assuming that lemma holds for $s = \alpha_0 \alpha_1 \ldots \alpha_{n-1}$ ($n > K$), we prove it holds for $s = \alpha_0 \alpha_1 \ldots \alpha_n$. Recall that m_n is the state in $G_{K,obs}$ that is reachable with s (which in the Lemma is denoted by m). By construction we have

$$\begin{aligned}
m_n &= m_{n-1} \| M(\alpha_n) \\
&= \{(i_1, \ldots, i_{K+1}) | (i_0, i_1, \ldots, i_K) \in m_{n-1}, (i_K, i_{K+1}) \in M(\alpha_n)\} \quad \text{(Definition of } \| \text{ operator)} \\
&= \{(i_1, \ldots, i_{K+1}) | (0 \leq k \leq K+1, 0 \leq l \leq K-1) : i_k \in X, \exists t_l \in \Sigma^*, P(t_l) = \alpha_{n-K+l}, \\
&\quad \delta(i_l, t_l) = i_{l+1}, \exists j \in X, t \in \Sigma^*, \delta(j, t) = i_0, P(t) = \alpha_0 \alpha_1 \ldots \alpha_{n-K-1}, \\
&\quad (i_K, i_{K+1}) \in M(\alpha_n)\} \quad \text{(by induction)} \\
&= \{(i_1, \ldots, i_{K+1}) | (0 \leq k \leq K+1, 0 \leq l \leq K-1) : i_k \in X, \exists t_l \in \Sigma^*, P(t_l) = \alpha_{n-K+l}, \\
&\quad \delta(i_l, t_l) = i_{l+1}, \exists j \in X, t \in \Sigma^*, \delta(j, t) = i_0, P(t) = \alpha_0 \alpha_1 \ldots \alpha_{n-K-1}, \exists t_K \in \Sigma^*, \\
&\quad \delta(i_K, t_K) = i_{K+1}, P(t_K) = \alpha_n\} \text{ (by definition of } M(\alpha_n)) \\
&= \{(i_1, \ldots, i_{K+1}) | (1 \leq k \leq K+1, 1 \leq l \leq K) : i_k \in X, \exists t_l \in \Sigma^*, P(t_l) = \alpha_{n-K+1+l}, \delta(i_l, t_l) = i_{l+1}, \\
&\quad \exists j \in X, t' \in \Sigma^*, \delta(j, t') = i_1, P(t') = \alpha_0 \alpha_1 \ldots \alpha_{n-K}\}
\end{aligned}$$

Note that in the last line, we define $t' = tt_1$ from the previous line. The proof is completed by decreasing the indices by one. □

Next we prove that the K-delay state estimator captures the K-delayed state estimates as the starting state of the trellis mapping stored in its state (similar to Corollary 1).

Corollary 2. *The K-delayed state estimate $\hat{X}_{|s|-K}(s)$ after observing s can be captured using the K-delay state estimator as follows: suppose $\delta_{K,obs}(X_{K,0}, s) = m$, then*

$$\hat{X}_{|s|-K}(s) = m(K).$$

Proof. Since the K-delayed state estimate is only defined for $|s| \geq K$, by Lemma 1 we have

$$\begin{aligned} m &= \delta_{K,obs}(X_{K,0}, s) \\ &= \{(i_0, i_1, \ldots, i_K) | (0 \leq k \leq K, 0 \leq l \leq K-1) : i_k \in X, \exists t_l \in \Sigma^*, P(t_l) = \alpha_{n-K+l+1}, \\ &\quad \delta(i_l, t_l) = i_{l+1}, \exists j \in X, t \in \Sigma^*, \delta(j, t) = i_0, P(t) = \alpha_0\alpha_1 \ldots \alpha_{n-K}\}. \end{aligned}$$

Moreover,

$$\begin{aligned} m(K) &= \{i_0 | (i_0, i_1, \ldots, i_K) \in m\} \\ &= \{i_0 | (0 \leq k \leq K, 0 \leq l \leq K-1) : i_k \in X, \exists t_l \in \Sigma^*, P(t_l) = \alpha_{n-K+l+1}, \delta(i_l, t_l) = i_{l+1}, \\ &\quad \exists j \in X, t \in \Sigma^*, \delta(j, t) = i_0, P(t) = \alpha_0\alpha_1 \ldots \alpha_{n-K}\} \\ &= \{i_0 | \exists s', s'' \in \Sigma^*, \exists j \in X : \delta(j, s') = i_0, \delta(i_0, s'') \text{ is defined}, \\ &\quad P(s') = \alpha_0\alpha_1 \ldots \alpha_{n-K}, P(s'') = \alpha_{n-K+1} \ldots \alpha_n\} \\ &= \hat{X}_{|s|-K}(s), \end{aligned}$$

where the third equation follows from the second equation with $s'' = t_0t_1 \ldots t_{K-1}$ and $s' = t$. This completes the proof. □

We demonstrate the construction of the K-delay state estimator via the following example.

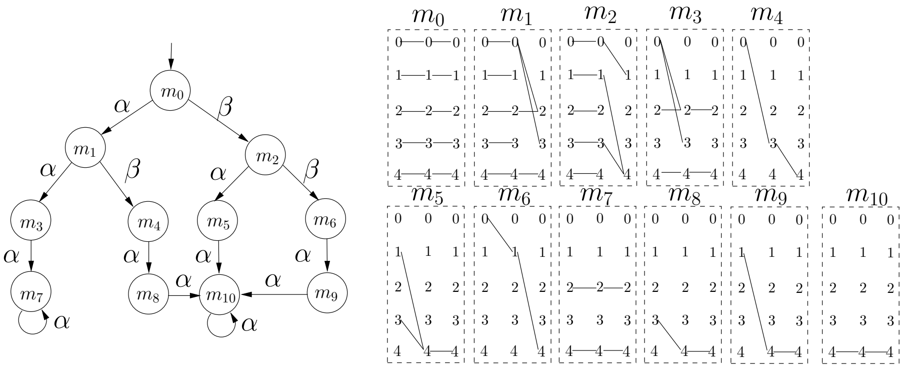


Figure 6: 2-delay state estimator corresponding to DES G discussed in Example 6.

Example 6. *Consider the DES G in Figure 1-a. For this system, the 2-delay state estimator is represented in Figure 6 along with the trellis mappings $m_0, m_1, \ldots, m_{10}$ needed in the construction. The initial uncertainty is the whole state space and the initial state of the state estimator captures this in m_0 via a trellis mapping that maps each system state to itself as*

$$\begin{aligned} X \odot_2 X &= \{(0,0,0),(1,1,1),(2,2,2),(3,3,3),(4,4,4)\} \\ &\equiv m_0. \end{aligned}$$

Starting from the initial state and upon observing α, the next state in the 2-delay state estimator m' can be constructed following Definition 8 as

$$\begin{aligned} m' &= \delta_{K,obs}(m_0, \alpha) \\ &= m_0 \| M(\alpha) \\ &= \Big\{(0,0,0),(1,1,1),(2,2,2),(3,3,3),(4,4,4)\Big\} \| M(\alpha) \\ &= \Big\{(0,0,0),(1,1,1),(2,2,2),(3,3,3)\Big\} \| \Big\{(0,2),(0,3),(2,2),(4,4)\Big\} \\ &= \{(0,0,2),(0,0,3),(2,2,2),(4,4,4)\} \\ &\equiv m_1. \end{aligned}$$

Example 2 explains how $M(\alpha)$ is synthesized. Note that on the right of Figure 6 we use trellis diagrams to describe the triples associated with the 2-delay state estimator. Next, assume that *we observe β which results in $M(\beta) = \{(1,0),(1,1),(1,2),(3,1),(3,3)\}$. Based on the notation in Definition 8, we have*

$$\begin{aligned} m' &= \delta_{K,obs}(m_1, \beta) \\ &= m_1 \| M(\beta) \\ &= \{(0,0,2),(0,0,3),(2,2,2),(4,4,4)\} \| \{(1,0),(1,1),(1,2),(3,1),(3,3)\} \\ &= \{(0,3,4)\} \\ &\equiv m_4. \end{aligned}$$

This implies that $\alpha\beta$ can only be observed if the system follows the state trajectory $0 \to 3 \to 4$. Using this approach for all possible observations (from each state), the 2-delay state estimator construction can be completed as shown in Figure 6. □

Remark 4. *Note that the K-delay state estimator captures the trellis mapping resulting from the last K observations, i.e., the trellis mappings summarize a constant size moving window of observations. On the other hand, in the ISE the state mappings summarize all the observations made so far (i.e., the size of the window of the observations is increasing), but only keep information about the starting and final states (and the connections between them).*

5 Analysis of Opacity Notions in DES

In this section, we use the delayed state estimators we developed in Section 4 to address the problem of verifying the opacity notions introduced in Section 3.

5.1 Verifying Initial-State Opacity

To verify initial-state opacity, we model the intruder as an initial-state estimator (ISE) for the system and check whether, at any point during the observation sequence, the estimate of the initial state falls completely within the secret set. We formalize this using the following theorem.

Theorem 2. *Discrete event system G is (S, P, ∞) initial-state opaque if and only if*

$$\forall m \in X_{\infty,obs} : m(1) \cap 2^S = \emptyset,$$

where $X_{\infty,obs}$ is the set of states in $G_{\infty,obs}$ that is reachable from the initial state $X_{\infty,0} = X \odot X$.

Proof. The condition in the theorem is equivalent to

$$\forall m \in X_{\infty,obs} : \ (j,k) \in m, j \in S \Rightarrow \exists (j', k') \in m, j' \in X - S,$$

which implies that there is no point along the observation such that the estimate of the initial state is entirely within secret state; in other words, the condition in the theorem is equivalent to the fact that the system is (S, P, ∞) initial-state opaque. □

Example 7. *In this example we show that DES G in Figure 1-a is not $(\{0\}, P, \infty)$ initial-state opaque. Consider the initial-state estimator depicted in Figure 5. State m_4 in the ISE, which can be reached via sequences of the form $\alpha\beta\alpha^*$ violates $(\{0\}, P, \infty)$ initial-state opacity since its set of starting states is $\{0\}$ which is within the set of secret states S. In other words, observing $\alpha\beta\alpha^*$ completely determines the initial state as state 0 which is within the set of secret states (and hence violates initial-state opacity). This system, however, is $(\{3\}, P, \infty)$ initial-state opaque. State mappings m_2 and m_5 contain state 3 in their set of starting states. This does not violate initial-state opacity since the sets of starting states of both of these state mappings also contain some states outside the set of secret states: $m_2(1) = \{0, 1, 3\}$ and $m_5(1) = \{1, 3\}$.* □

Remark 5. *Note that if partial knowledge of the initial state is available, the notion of initial state opacity and the construction of the ISE can be modified accordingly (refer to Remark 1).*

5.2 Verifying K-Step Opacity

In Section 4 we proved that the set of starting states of the trellis mappings in the K-delay state estimator capture the estimates of the system states K observations ago (i.e., after having an opportunity to observe K events past the point when the system was at that particular state).

Using this fact, we show that the K-delay state estimator can be used for modeling the intruder and hence for verifying K-step opacity. Specifically, a DES G is K-step opaque if and only if all the starting states associated with the trellis mappings in the reachable states of the K-delay state estimator contain at least one element outside S. The following theorem states this formally.

Theorem 3. *Discrete event system G is (S, P, K)-opaque if and only if*

$$\forall m \in X_{K,obs}, k \in \{0, \dots, K\} : m(k) \cap 2^S = \emptyset,$$

where $X_{K,obs}$ is the set of states in $G_{K,obs}$ that is reachable from the initial state $X_{K,0} = X \odot_K X$.

Proof. (If) Proof by contradiction. Without loss of generality assume that there exists $m \in X_{K,obs}$ reached via $t = \alpha_0 \dots \alpha_n$ such that $m(K) \cap 2^S \neq \emptyset$. Using Corollary 2, we have

$$\begin{aligned} m(K) &= \hat{X}_{|t|-K}(t) \\ &= \{i_0 | \exists t', t'' \in \Sigma^*, \exists j \in X : \delta(j, t') = i_0, \delta(i_0, t'') \text{ is defined}, \\ &\quad P(t') = \alpha_0 \alpha_1 \dots \alpha_{n-K}, P(t'') = \alpha_{n-K+1} \dots \alpha_n\}. \end{aligned}$$

Hence $m(K) \cap 2^S \neq \emptyset$ implies that

$$\exists t \in \Sigma^*, \exists t' \in \bar{t}, |P(t)/P(t')| = K, \forall j \in X, \delta(j, t) \text{ is defined} : \delta(j, t') \in S.$$

Hence, we find a string t such that it passes through the set of secret states exactly K observations ago and such that there is no other sequence in the system that creates the same output and passes through a non-secret state K observations ago. Therefore, the system is not K-step opaque. Similarly, we can prove that $m(k) \cap 2^S \neq \emptyset$ for $k \in \{1, \dots, K\}$ implies that system is not K-step opaque.

(Only if) Assume that system is not K-step opaque. Hence,

$$\begin{aligned} &\exists t \in \Sigma^*, \exists t' \in \bar{t}, |P(t)/P(t')| \leq K, \forall j \in X, \delta(j, t) \text{ is defined} : \Big(\delta(j, t') \in S, \nexists s \in \Sigma^* : \\ &\qquad \exists i' \in X, \exists s' \in \bar{s}, P(s) = P(t), P(s') = P(t'), \delta(i', s') \notin S, \delta(i, s) \text{ is defined}\Big). \end{aligned} \tag{3}$$

Assume that $P(t) = \alpha_0\alpha_1 \ldots \alpha_n$ and $P(t') = \alpha_0\alpha_1 \ldots \alpha_{n-k}$ for some $k < K$. Now consider the state m that is reached via $P(t)$ in $G_{K,obs}$. From Lemma 1 we have that

$$m = \{(i_0, i_1, \ldots, i_K)|(0 \leq k \leq K, 0 \leq l \leq K-1) : i_k \in X, \exists t_l \in \Sigma^*, P(t_l) = \alpha_{n-K+1+l},$$

$$\delta(i_l, t_l) = i_{l+1}, \exists j \in X, \omega \in \Sigma^*, \delta(j, \omega) = i_0, P(\omega) = \alpha_0\alpha_1 \ldots \alpha_{n-K}\}.$$

Hence,

$$m(k) = \{i|(K-k \leq l \leq K-1) : i \in X, \exists t_l \in \Sigma^*, P(t_l) = \alpha_{n-K+1+l},$$

$$\delta(i_l, t_l) = i_{l+1}, \exists j \in X, t' \in \Sigma^*, \delta(j, t') = i, P(t') = \alpha_0\alpha_1 \ldots \alpha_{n-k}\}.$$

Now since the system is not opaque, (3) implies that

$$\forall i \in m(k) : i \in S.$$

Hence there exists $0 \leq k \leq K$ such that

$$m(k) \cap 2^s \neq \emptyset.$$

This completes the proof. □

The following example clarifies the above points.

Example 8. *In this example we show that DES G in Figure 1-a is $(\{0\}, P, 0)$-opaque and $(\{0\}, P, 1)$-opaque but not $(\{0\}, P, 2)$-opaque. Consider the state estimator with zero delay for this system depicted in Figure 1-b. The only state that contains the secret state is the initial state X which also contains states outside the set of secret state $\{0\}$. Hence, DES G is $(\{0\}, P, 0)$-opaque. A similar argument can be made for $(\{0\}, P, 1)$-opacity (by constructing the 1-delay state estimator). DES G is not $(\{0\}, P, 2)$-opaque due to the existence of state m_4 and m_6 in the 2-delay state estimator depicted in Figure 6. If the system generates the sequence of observations $\alpha\beta$ or $\beta\beta$, then (since the only state from which $\beta\alpha$ or $\beta\beta$ can be observed is state 0) we can conclude with certainty that the system was in state 0 two steps ago. This violates the 2-step opacity requirement since state 0* *is a secret state. Note that this example demonstrates that K-step opacity does not imply K'-step opacity for $K' > K$.* □

5.3 Verifying Infinite-Step Opacity

As mentioned earlier, infinite-step opacity can be considered as the limiting case of K-step opacity as $K \to \infty$. Note that in this case, the K-delay state estimator is not a finite structure anymore (because $K \to \infty$) and hence it is not useful in modeling the intruder or verifying infinite-step opacity. In this section, we introduce a new method for verifying this property using both the ISE and the zero-delay state estimator.

To verify that a system is infinite-step opaque, we need to verify that at any point along the observation, knowing the observations that have been made before reaching that point *in addition* to all possible future observations (from that point onward) does not (and/or will not) allow us to determine whether the set of possible states at that point is a subset of the set of secret states. Hence, the verification consists of two phases: (i) find all possible estimates of the system state along any possible sequence of observations, and (ii) for each point in this trajectory (set of possible system states), calculate all possible information that can be gained about that point from future observations from that point onward. The first phase can be achieved via a standard zero-delay 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. In other words, for each set of estimates $Z \subseteq X$ provided in the first phase, we construct an ISE whose initial state is associated with the state mapping $\{(z, z)|z \in Z\}$. Clearly, if any of these ISEs contains one state mapping associated with an estimate of the initial state whose set of starting state contains elements only in S, then DES G is not infinite-step opaque. The following theorem formalizes this discussion.

Theorem 4. *For every set of state estimates Z_n associated with a state of $G_{0,obs}$, construct $G^n_{\infty,obs}$ by setting its initial state $X^n_{\infty,0}$ to be $\{(z, z)|z \in Z_n\}$. Then, DES G is (S, P, ∞)-opaque if and only* *if*

$$\forall n, \forall m \in X^n_{\infty,obs} : m(1) \cap 2^S = \emptyset, \tag{4}$$

where $X^n_{\infty,obs}$ is the set of states in $G^n_{\infty,obs}$ that is reachable from initial state $X^n_{\infty,0} = \{(z,z)|z \in Z_n\}$.

Proof. (If) We prove this by contradiction. Assume that DES G is not (S,P,∞)-opaque. Hence, there exists a string t that passes through a secret state j such that every other string s with $P(s) = P(t)$ also passes through a secret state j' when string t passes through the secret state j. Without loss of generality assume that there is only one such string s. (Refer to Figure 2 for graphical representation.) Suppose that string t (string s) originates from state i (state i') and that string t (string s) can be written as $t = t't''$ ($s = s's''$) such that state i (state i') reaches state j (state j') via string t' (string s'), and state j (state j') reaches some state k (state k') via string t'' (string s''). Note that string s visits state j' when string t visits state j means that $P(s') = P(t')$. Since state j and state j' can be reached from state i and state i' via strings s' and t' with the same projection ($P(s') = P(t')$), the properties of the zero-delay state estimator [6] imply that $\{j,j'\} \in Z_n$ for some Z_n associated with states of $G_{0,obs}$. By construction of $G^n_{\infty,obs}$, this implies that $(j,j),(j',j') \in X^n_{\infty,0}$. Consider the state m that is reached in $G^n_{\infty,obs}$ from $X^n_{\infty,0}$ via $P(t'')$ (which equals $P(s'')$ since $s = s's''$, $t = t't''$, $P(s) = P(t)$ and $P(s') = P(t')$). Since state j (state j') reaches state k (state k') via string t'' (string s''), by Lemma 1, $m = \{(j,k),(j',k')\}$. Moreover (4) implies that $\{j,j'\} \cap 2^S = \emptyset$ which is a contradiction since we assumed $\{j,j'\} \in S$. This completes the (If) part of the proof.

(Only if) Assume that DES G is (S,P,∞)-opaque. We need to prove that for all n and $m \in X^n_{\infty,obs}$, $(j,k) \in m$ and $j \in S$ implies that there exists $(j',k') \in m$ such that $j' \in X - S$. We prove this by contradiction. Assume that there exists n and $m \in X^n_{\infty,obs}$ such that for all $(j,k) \in m$ we have $j \in S$, and m is reached via ω in $G^n_{\infty,obs}$. Without loss of generality suppose that $m = \{(j,k),(j',k')\}$. Using Lemma 1 and following the notations used in Figure 2 we have for all $t'', s'' \in \Sigma^*$ such that $P(t'') = P(s'') = \omega$, $\delta(j,t'') = k$, and $\delta(j',s'') = k'$, then $\{j,j'\} \subseteq S$. As mentioned before, the initial state $X^n_{\infty,0}$ of $G^n_{\infty,obs}$ is constructed using an estimate of the current state Z which is a reachable state in $G_{0,obs}$. Assume that Z is reachable in $G_{0,obs}$ via a string Ω from the initial state X_0. By construction of $G_{0,obs}$, we know that there exists $i,i' \in X$, and $t',s' \in \Sigma^*$ such that $\delta(i,t') = j$,

$\delta(i', s') = j'$, and $P(s') = P(t') = \Omega$. Define $t = t't''$ and $s = s's''$ (refer to Figure 2) and assume that we observe $\Omega\omega$. For this observation, there is a string t such that a prefix t' of t passes through the secret state j and every other string s that has the same projection as t, i.e., $P(s) = P(t)$, also passes through the set of secret states when t does (i.e., at the time $P(t')$ is observed). This violates infinite-step opacity which is a contradiction and hence the proof is complete. □

Remark 6. *Note that in practice, since the set of initial state estimates can only decrease with additional observations, we only need to consider $G^n_{\infty,obs}$ for Z_n's which have a nonzero intersection with S.*

Example 9. *In this example, we verify that DES G in Figure 1-a is not $(\{3\}, P, \infty)$-opaque. To verify infinite-step opacity we need to first construct the zero-delay state estimator $G_{0,obs}$ as in Figure 1-b. As mentioned in Example 6, this state estimator has five states $\{\{4\}, \{1,4\}, \{2,4\}, \{2,3,4\}, \{0,1,2,3,4\}\}$ and 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 (i.e., state mappings) only state mapping $\{(0,0),(1,1),\ (2,2), (3,3),\ (4,4)\}$ which corresponds to $Z_1 = \{0,1,2,3,4\}$ and state mapping $\{(2,2),(3,3),(4,4)\}$ which corresponds to state $Z_2 = \{2,3,4\}$ contain the secret state 3. By Remark 6, this implies that we only need to construct two ISEs corresponding to these two state mappings. (i) The ISE $G^1_{\infty,obs}$ with initial state mapping corresponding to Z_1 is indeed the initial-state estimator we considered previously in Example 5 and is depicted in Figure 5. 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 state. (ii) The ISE $G^2_{\infty,obs}$ with initial state corresponding to Z_2 is depicted in Figure 7. State $m_2 = \{(3,4)\}$ in $G^2_{\infty,obs}$ 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 $G^2_{\infty,obs}$ via β from m_0. Moreover m_0 in this ISE corresponds to the state in $G_{0,obs}$ (in Figure 1-b) that was reached via observation α. Putting these two pieces of information together, we conclude that observing $\alpha\beta$ reveals that system has gone through state 3, which is a secret state; hence the system is not $(\{3\}, P, \infty)$-opaque.* □

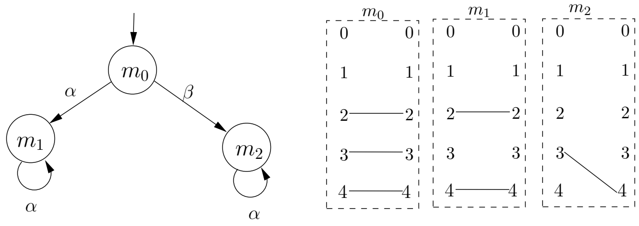


Figure 7: Initial-state estimator $G^2_{\infty,obs}$ corresponding to state $Z_2 = \{2, 3, 4\}$ in the zero-delay state estimator of the DES G in Figure 1.

5.4 Role of Delay in K-Step Opacity

In this section we show that for $K \geq 2^{N^2} - 1$, K-step opacity implies infinite-step opacity. Specifically, we show that for $K' > K \geq 2^{N^2} - 1$, K-step opacity and K'-step opacity are equivalent and, since infinite-step opacity is the limiting case of K-step opacity as $K \to \infty$, we can state that K-step opacity for $K \geq 2^{N^2} - 1$ is equivalent to infinite-step opacity (the other direction is obviously true: infinite-step opacity always implies K-step opacity). Note that K-step opacity does not in general imply K'-step opacity for $K' > K$ (in fact, Example **??** is a demonstration of this).

The idea behind the proof is the following: fix a point in the system's state trajectory. In the K-step opacity problem we are interested in finding how much we can say regarding the membership of the state, at that fixed point in time, within the set of secret states, even after we make K additional observations. More precisely, we need to determine if at any point in time (at this fixed point in time or in the future) we can unambiguously determine that the state at this fixed point falls within the set of secret states. We can gain insight to this question by considering the estimate of the state at that fixed point as the initial uncertainty for an ISE. Observe that since the ISE has at most 2^{N^2} states (because there are that many different state mappings), we are guaranteed that each state in that ISE is reachable via a string that generates at most K observations as long as $K \geq 2^{N^2} - 1$. Note that here we are concerned with the information conveyed by the *set* of all sequences of observations of length at most K, and not a specific sequence of observations. Observe that sequences that involve cycles in the ISE can be excluded because they do not lead to new states; equivalently, each one of them behaves identically to a sequence without cycles which has length at most K. For this reason, there would be no more information if observation sequences of length more than K were included in this set.

We carry the formal proof of the above statement in two theorems. First, in Theorem 5, we prove that for the aforementioned fixed point in the state trajectory, for $K \geq 2^{N^2} - 1$, the information regarding the state (at the same point in time) for all K-delayed state estimates is equivalent. Then, in Theorem 6, we show that this equivalence results in the equivalence of K-step opacity and K'-step opacity for $K' > K \geq 2^{N^2} - 1$.

Theorem 5. *Assume that $G_{K,obs}$ and $G_{K^*,obs}$ with $K > K^* = 2^{N^2} - 1$ are delayed state estimators for a finite automaton $G = (X, \Sigma, \delta)$ and are constructed as described in Section 2. Then, for any trellis mapping m reached in $G_{K,obs}$ via $s = \alpha_0\alpha_1 \dots \alpha_n$ for $|s| > 2^{N^2} - 1$ and for each $m(k)$, $2^{N^2} \leq k \leq K$, there exists a trellis mapping m' associated with a state reached in $G_{K^*,obs}$ via some $s' = \alpha_0\alpha_1 \dots \alpha_{n-k}\alpha'_{n-k+1} \dots \alpha'_{n'}$ for some $n' \leq n + 2^{N^2} - 1 - k$ and with $\alpha'_{n-p} \in \Sigma_{obs}, k-1 \leq p \leq n - n'$, such that $m(k) = m'(l)$, for $l = k + n' - n$.*

Proof. Recall that in any K-delay state estimator $G_{K,obs}$, the k-delayed state estimate due to observation s is captured via the set $m(k)$ where m is the trellis mapping associated with the state reached in $G_{K,obs}$ via s (k satisfies $0 \leq k \leq K$). Now consider the fixed point in time after the sequence of observations $\alpha_0\alpha_1 \dots \alpha_{n-k}$ has been observed. Assume that k more observations have been made since the system passed through the state at that fixed point in time, i.e., $\alpha_{n-k+1}\alpha_{n-k+2} \dots \alpha_n$ are observed. Then, the set $m(k)$ denotes the k-delayed state estimate at that fixed point due to the sequence of observations $s = \alpha_0\alpha_1 \dots \alpha_{n-k}\alpha_{n-k+1} \dots \alpha_n$. Similarly, $m'(l)$ denotes the l-delayed state estimates of that fixed point due to the sequence of observations $s' = \alpha_0\alpha_1 \dots \alpha_{n-k}\alpha'_{n-k+1} \dots \alpha'_{n'}$ for $n' - l = n - k$. In other words, $m(k)$ represents the k-delayed state estimate, if after the passage of the system through the state at that fixed point $\alpha_{n-k+1}\alpha_{n-k+2} \dots \alpha_n$ has been observed, whereas $m'(l)$ denotes the l-delayed state estimate at this same point if $\alpha'_{n-k+1} \dots \alpha'_{n'}$ has been observed since the passage time. To prove Theorem 5, we need to show that assuming $k \geq 2^{N^2}$, for the k-delayed state estimate of that fixed point due to the sequence of observations $s = \alpha_0\alpha_1 \dots \alpha_{n-k}\alpha_{n-k+1} \dots \alpha_n$, there is an $l \leq 2^{N^2} - 1$ such that the l-delayed state estimate of that same fixed point due to a shorter sequence of observations $s' = \alpha_0\alpha_1 \dots \alpha_{n-k}\alpha'_{n-k+1} \dots \alpha'_{n'}$ where $n' = l + n - k$ is the same. Denote the estimate of the system's (current) state at that point (i.e., the estimate after observing $\alpha_0\alpha_1 \dots \alpha_{n-k}$) by $Z \subseteq X$. The problem of k-delayed estimation of the state of the system at the fixed point in time after observing $s = \alpha_0\alpha_1 \dots \alpha_n$ can be considered as an initial state estimation problem where (due to the observations $\alpha_0\alpha_1 \dots \alpha_{n-k}$ that have been made before reaching that fixed point) the initial uncertainty about the "initial state" is the set Z. Hence, the set $m(k)$ after observing $s = \alpha_0\alpha_1 \dots \alpha_n$ is the same as the set of starting states of the state mapping that is associated with the state that is reached via $\alpha_{n-k+1}\alpha_{n-k+2} \dots \alpha_n$ in the ISE whose initial state is $\{(z,z)|z \in Z\}$ (and, of course, Z is the set of state estimates given by a zero-delay (standard) state estimator after observing $\alpha_0\alpha_1 \dots \alpha_{n-k}$). Note that this string has length $k > 2^{N^2} - 1$. Since the ISE has at most 2^{N^2} states, strings of length at most $2^{N^2} - 1$ can be chosen to visit any state of the ISE. This implies that the state reached in the modified ISE via the string $\alpha_{n-k+1}\alpha_{n-k+2} \dots \alpha_n$ of length k can also be reached via a string of length less than or equal to $2^{N^2} - 1$, which we denote by $\alpha'_{n-k+1}\alpha'_{n-k+2} \dots \alpha'_{n'}$ for some $n' \leq n + 2^{N^2} - 1 - k$. Since the states reached via both of these strings in this ISE are the same, the k-delayed state estimate due to s is the same as the $(k-(n-n'))$-delayed state estimate due to s'. Letting $(k-(n-n')) = l$ completes the proof. □

The above result can be used to show that K'-step opacity is equivalent to K-step opacity for $K' > K \geq 2^{N^2} - 1$. We prove this by showing that for $K \geq 2^{N^2}$, K-step opacity is equivalent to K^*-step opacity with $K^* = 2^{N^2} - 1$.

Theorem 6. *For a finite automaton $G = (X, \Sigma, \delta)$, K-step opacity is equivalent to K^*-step opacity for $K > K^* = 2^{N^2} - 1$ where $N = |X|$.*

Proof. (K-step opacity ⇒ K^*-step opacity) Recall that DES is K-step opaque if and only if

$$\forall m \in X_{K,obs} : m(k) \cap 2^S = \emptyset, 0 \leq k \leq K. \tag{5}$$

Consider the trellis mappings m and m' associated with the state reached in $G_{K,obs}$ and $G_{K^*,obs}$ via s. Observe that $m(k) = m'(k), 0 \leq k \leq 2^{N^2} - 1$; since both $m(k)$ and $m'(k)$ denote the k-delayed state estimate due to observation s, they are identical set of states. Therefore, (5) implies that $\forall m \in X_{K^*,obs} : m(k) \cap 2^S = \emptyset, 0 \leq k \leq K^*$, which implies that K^*-step opacity holds.

(K^*-step opacity ⇒ K-step opacity) We need to show (5). From Theorem 5 we have: for any trellis mapping m associated with states of $G_{K,obs}$ reached via a string s with $|s| \geq 2^{N^2} - 1$ and $2^{N^2} \leq k \leq K$, there is a trellis mapping m' associated with states of $G_{K^*,obs}$ and some l satisfying $0 \leq l \leq 2^{N^2} - 1$ such that $m(k) = m'(l)$. Now if DES G is K^*-step opaque then all sets of intermediate states $m'(l)$ of any trellis mapping m' associated with states in $G_{K^*,obs}$ contain states outside the set of secret states; following the previous discussion, for $2^{N^2} \leq k \leq K$, all set of intermediate states $m'(k)$ of any trellis mapping m' associated with states of $G_{K,obs}$ contain states outside the set of secret states. This implies (5) for $2^{N^2} \leq k \leq K$. Moreover, the discussion in part (i) implies (5) for $0 \leq k \leq 2^{N^2} - 1$. Therefore (5) holds if m is reached in $G_{K,obs}$ via a string s with $|s| \geq 2^{N^2} - 1$. If m is reached via a shorter string t with $|t| < 2^{N^2} - 1$, then the discussion in part (i) still implies (5) for $0 \leq k < 2^{N^2} - 1$; moreover for $2^{N^2} - 1 \leq k \leq K$, we have $m(k) \equiv X$ (since we have yet to make enough observations) which trivially satisfies (5). □

5.5 Remarks

We now consider how three notions of opacity are related. Initial-state opacity is different from (S, P, K)-opacity since in initial-state opacity, we fix a point in time (initial point) and require that the intruder cannot infer the membership of the state at that point to the set of secret states, regardless of the length of the observation sequence. On the other hand, in (S, P, K)-opacity we require this to be true for all the states covered in a window of size K around the current observation.

Initial-state opacity is similar to (S, P, ∞)-opacity in the sense that both require opacity for the *whole* length of the observation. However, they are different since in initial-state opacity, we only require opacity of a fixed point in the state trajectory whereas in (S, P, ∞), this is required for all the states visited along the state trajectory. In other words, initial-state opacity is not violated if, during the observation, the intruder can infer that the current state is within the set of secret states as long as the intruder cannot locate the initial state within the secret state. As a result, initial state-opacity does not imply infinite-step opacity. Note that infinite-step opacity implies the other two opacities.

The final point is that, as mentioned in Theorem 6, K-step opacity implies K'-step opacity for $K' > K \geq 2^{N^2} - 1$ (where N is the number of states) and, since infinite-step opacity is the limiting case of K-step opacity as $K \to \infty$, we can use the same bound $(2^{N^2} - 1)$ to state that K-step opacity for $K \geq 2^{N^2} - 1$ is equivalent to infinite-step opacity (since infinite-step opacity always implies K-step opacity).

6 Conclusion

In this report, we addressed the problem of delayed estimation in discrete event systems that are modeled by finite automata with partial observations of their transitions. Specifically, we investigated the estimation of the initial state of the system and the estimation of the state of the system at a fixed number of observations into the past: we provided two state estimators that capture these estimates and showed that the information included in the delay state estimators does not increase by increasing delay if the current delay is bigger than $2^{N^2} - 1$ where N is the number of states.

We also demonstrated the application of delayed estimation in characterizing security properties (and hence the application of delay state estimators in verifying these properties). For this, we extended three notions of opacity: initial-state opacity, K-step opacity and infinite-step opacity. These notions are state-based notions which require opacity of the membership of the system states to a secret set of states. To show the relevance of the delay state estimator, we used our two state estimators to model an intruder/observer of the given system, and used these state estimators to obtain verification methods for these security notions. We also proved a delay K under which infinite-step opacity and K-step opacity are equivalent.

In our future work we plan to use techniques from supervisory control [17] to design minimally restrictive supervisors for a given discrete event system in a way that it ensures certain desirable security properties.

References

[1] S. Bose, A. Patra, and S. Mukhopadhyay, “On observability with delay: antitheses and syntheses,” *IEEE Transactions on Automatic Control*, vol. 39, no. 4, pp. 803–806, April 1994.

[2] P. E. Caines, R. Greiner, and S. Wang, "Dynamical logic observers for finite automata," in *Proc. of 27th IEEE Conference on Decision and Control (CDC 1988)*, vol. 1, December 1988, pp. 226–233.

[3] S. Hashtrudi Zad, H. Kwong, and W. Wonham, "Fault diagnosis in discrete event systems: Framework and model reduction," *IEEE Transactions on Automatic Control*, vol. 48, no. 7, pp. 1199–1212, July 2003.

[4] F. Lin and W. Wonham, "On observability of discrete event systems," *Information Sciences*, vol. 44, no. 3, pp. 173–198, April 1988.

[5] M. Oishi, I. Hwang, and C. Tomlin, "Immediate observability of discrete event systems with application to user–interface design," in *Proc. of 42nd IEEE Conference on Decision and Control (CDC 2003)*, vol. 3, December 2003, pp. 2665–2672.

[6] C. M. Özveren and A. S. Willsky, "Observability of discrete event dynamic systems," *IEEE Transactions on Automatic Control*, vol. 35, no. 7, pp. 797–806, July 1990.

[7] P. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event processes," *SIAM Journal on Control and Optimization*, vol. 25, no. 1, pp. 206–230, January 1987.

[8] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. C. Teneketzis, "Failure diagnosis using discrete event models," *IEEE Transactions on Automatic Control*, vol. 4, no. 2, pp. 105–124, March 1996.

[9] C. M. Özveren, A. S. Willsky, and P. J. Antsaklis, "Stability and stabilizability of discrete event dynamic systems," *Journal of the Association for Computing Machinery*, vol. 38, no. 3, pp. 729–751, July 1991.

[10] C. M. Özveren and A. S. Willsky, "Invertibility of discrete event dynamic systems," *Mathematics of Control, Signals, and Systems*, vol. 5, no. 4, pp. 365–390, July 1992.

[11] Y. Li and W. M. Wonham, "Control of vector discrete event systems (I): The base model," *IEEE Transactions on Automatic Control*, vol. 38, no. 8, pp. 1214–1227, August 1993.

[12] B. H. Krogh and L. E. Holloway, “Synthesis of feedback control logic for discrete manufacturing systems,” *Automatica*, vol. 27, no. 4, pp. 641–651, July 1991.

[13] C. M. Özveren and A. S. Willsky, “Output stabilizability of discrete event dynamic systems,” in *Proc. of 28th IEEE Conference on Decision and Control (CDC 1989)*, vol. 3, September 1989, pp. 2719–2724.

[14] J. Bryans, M. Koutny, L. Mazare, and P. Ryan, “Opacity generalised to transition systems,” in *Proc. of the 3rd International Workshop on Formal Aspects in Security and Trust*, July 2005, pp. 81–95.

[15] A. Saboori and C. N. Hadjicostis, “Notions of security and opacity in discrete event systems,” in *Proc. of the 46th IEEE Conference on Decision and Control (CDC 2007)*, December 2007, pp. 5056–5061.

[16] C. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems.* Kluwer Academic Publishers, 1999.

[17] W. Wonham, *Supervisory Control of Discrete–Event Systems*, Systems Control Group, Department of Electrical and Computer Engineering, University of Toronto. Available at `www.utoronto.ca/DES`, 2005.

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

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

[20] D. K. Pradhan and M. Chatterjee, “GLFSR — a new test pattern generator for built–in–self–test,” *IEEE Transactions on Computer–Aided Design of Integrated Circuits and Systems*, vol. 18, no. 2, pp. 238–247, February 1999.

[21] R. Anderson, “On Fibonacci keystream generators,” in *Proc. of International Workshop on Fast Software Encryption*, December 1994, pp. 346–352.

[22] M. Briceno, I. Goldberg, and D. Wagner. A pedagogical implementation of the GSM A5/1 and A5/2 voice privacy encryption algorithms. [Online]. Available: http://www.scard.org/gsm/a51.html.

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

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

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