Technical Notes and Correspondence

Error- and Tamper-Tolerant State Estimation for Discrete Event Systems Under Cost Constraints

Yuting Li, Christoforos N. Hadjicostis, Fellow, IEEE, Naiqi Wu, Fellow, IEEE, and Zhiwu Li, Fellow, IEEE

Abstract—This article deals with the state estimation problem in discrete-event systems modeled with nondeterministic finite automata, partially observed via a sensor measuring unit whose measurements (reported observed symbols) may be vitiated by a malicious attacker. The attacks considered in this article include arbitrary deletions, insertions, or substitutions of observed symbols by taking into account a bounded number of attacks or, more generally, a total cost constraint (assuming that each deletion, insertion, or substitution bears a positive cost to the attacker). We provide a characterization of the sequences of symbols that match the received sequence of possibly corrupted symbols, and subsequently use them to recursively perform tamper-tolerant state estimation subject to cost constraints. Each step of the recursive state estimation procedure has complexity of $O(|X| |\Sigma| C)$, where $|X|$ ($|\Sigma|$) is the number of states (events) of the given finite automaton and $C$ is the maximum total cost that is allowed for all deletions, insertions, and substitutions.

Index Terms—Data tampering, discrete-event system (DES), information corruption, least-cost error sequence, state estimation.

I. INTRODUCTION

State estimation in continuous-time systems was initiated in the 1950s and has been extensively applied to a variety of areas of engineering and science [1]. The primary motivation for state estimation is to be able to perform analysis of the current state of a system under the conditions characterized by a streaming sequence of measurements. The state estimator has knowledge of both the model of a system and the conditions characterized by a streaming sequence of measurements. The state estimator is aware of both the model of a system and the way it generates observations (outputs). Under appropriate redundancy levels, it can eliminate the effects of bad or erroneous measurements (in some cases, even account for temporary loss of measurements) without significantly affecting the quality of estimated values [2].

The state estimation problem in discrete-event systems (DESs) is essential since typically state information cannot be directly obtained due to limited sensor availability. For example, state estimation is critical for control, supervision, and opacity verification and enforcement [11], [12], [13], [14]. The problem becomes challenging because of possible faulty observations (e.g., due to cyber attacks, malfunctioning sensors, erroneous communication transmissions, or synchronization issues during the transmission of information from different sensors) [7], [15], [16], [17], [18], [19].

This article addresses centralized state estimation in DESs under adversarial attacks that corrupt the sensor readings. Some related work has appeared in the context of sensor attacks that drive a controlled DES to unsafe or undesirable states by manipulating observation sequences [20], [21]. The work in this article is also related to some existing state estimation and security results in the area of DESs [15], [17], [22], [23], [24], [25]. For example, the problem of state estimation has received considerable attention in DESs, particularly for checking and verifying state-based properties, such as detectability and opacity [8], [9], [10], [11], [12], [13], [14]. A system is said to be detectable if an external observer will (eventually) know the exact state of the system along all or some trajectories of the system [9]. By considering probabilities on system transitions, additional information can be incorporated in the state estimation problem [8]. Similarly, the property of diagnosability holds for a given system if, under all possible activities that may occur in the system, an external observer can determine the occurrence of fault events, after a finite number of subsequent events take place. In particular, Athanasopoulos et al. [17] considered fault diagnosis under unreliable observations: transpositions, deletions, and insertions of output symbols are formally defined with probabilities captured by a probabilistic finite automaton. In [15], a supervisor of a plant under partial observation is constructed to overcome attacks, that are modeled by a set-valued map, representing all possibly corrupted strings with respect to each original string.

With the development of networked control systems, data exchanged among networked components may suffer communication errors or malicious attacks [16], [26], [27]. In the framework of DESs, three typical types of cyber attacks are considered, namely deletions, insertions, and substitutions. A deletion (substitution) attack is a natural strategy, in which a valid data transmission is maliciously deleted (substituted) such that a system may deviate from its expected behavior (if this symbol is a control input) or an outside observer may incorrectly estimate its activity (if this symbol is an output of the system) [15]. An insertion attack is an attack that inserts extraneous symbols and can have similar effects as above. An insertion attack can also be used to render certain resources of a system unavailable, e.g., an attacker sends a huge number of fabricated packets to a device, with the intention of dramatically consuming amounts of endpoint network bandwidth [28].

In order to overcome such corruptions, the strategy proposed in this article first considers how a sequence estimation unit can calculate a set of matching sequences based on the possibly tampered sequence.

Digital Object Identifier 10.1109/TAC.2023.3239590
received from the channel. We aim to choose, among all matching sequences, the ones that closely match the one received at the sequence estimation unit. Therefore, a cost-value notion is proposed, where a positive value is assigned to each type of attack; this value is inversely related to the likelihoods of different types of attack occurrences. Matching sequences with less costs are more likely to have occurred and can be used to obtain educated estimates of states (or faults) that may have occurred in the system. Note that the number of matching sequences and their lengths can be infinite due to the existence of deletions. Hence, in this article, an upper bound on the total cost is set to limit the number of sequences that match an observed sequence.

A special case of this setting is to assume that the total number of attacks is bounded (this case arises as a special case when attacks have equal costs).

It is important to point out the difference between our study and the works in [8], [9], [10] on detectability analysis. Even though these works also form state estimates, our research considers unreliable observations under costs and has to incorporate cost constraints for a system that generates observations that can be attacked. Our work also shares some similarities with the works investigating fault diagnosis under nondeterministic observations in [29] and [30], but differs intrinsically since it considers corruptions due to malicious attacks, which can cause deletions and substitutions of symbols, as well as insertions of sequences of symbols. As we will see, cost constraints on attacks (as considered in this article) can greatly affect the outcome of state estimation (and thus fault diagnosis and diagnosability) for a given system under unreliable observations.

The main contributions of this article are as follows:
1) It formulates and solves the tamper-tolerant state estimation problem under communication attacks of an unconstrained total cost or a bounded total cost, where each type of attack is associated with a positive integer cost.
2) It proposes appropriate transformations of a given system, for both cases of unconstrained and cost-constrained attacks; these transformations attack attacks and costs to different enhanced versions of the plant model and can be used to obtain recursive tamper-tolerant state estimation algorithms of polynomial complexity.
3) A Viterbi-like algorithm is developed to perform the least cost tamper-tolerant state estimation, where only the least-cost paths from each initial state to each reachable state are retained.

II. BACKGROUND AND PRELIMINARIES

Let $\Sigma$ be an alphabet with a finite set of distinct symbols (events) $\sigma, \beta, \ldots$. As usual, $\Sigma^*$ denotes the set of all finite symbol sequences over $\Sigma$, including the empty sequence $\epsilon$ (sequence with no symbols). A member of $\Sigma^*$ is said to be a string or trace, and a subset of $\Sigma^*$ is a language defined over $\Sigma$. The length of a string $s \in \Sigma^*$ is the number of symbols in $s$, denoted by $|s|$ (with $|\epsilon| = 0$). Given strings $s, t \in \Sigma^*$, the concatenation of strings $s$ and $t$ is defined as the string $st$. For a string $s \in \Sigma^*$, $t \in \Sigma^*$ is said to be a prefix of $s$, if $(\exists t' \in \Sigma^*) s = tt'$. Given a language $L \subseteq \Sigma^*$, $L$ denotes the prefix-closure of $L$, defined as $L = \{ t \in \Sigma^* | \exists t' \in \Sigma^* : t = tt' \}$. By a slight abuse of notation, for $\sigma \in \Sigma$ and $s \in \Sigma^*$, we write $s \preceq \sigma$ to represent the event that $\sigma$ is in $s$, i.e., $s = s'\sigma s''$ for some $s', s'' \in \Sigma^*$. The cardinality of a set $X$ is the number of elements contained in $X$, denoted by $|X|$.

Definition 1: A deterministic finite automaton (DFA), denoted by $G$, is a four-tuple $G = (X, \Sigma, \delta, x_0)$, where $X$ is the set of states, $\Sigma$ is the set of events, $\delta : X \times \Sigma \rightarrow X$ is the partial state transition function, and $x_0 \in X$ is the initial state.

For convenience, $\delta$ is extended from domain $X \times \Sigma$ to $X \times \Sigma^*$ in the standard recursive manner: $\delta(x, \epsilon) = x$; $\delta(x, \sigma s) = \delta(\delta(x, \sigma), s)$ for $x \in X$, $\sigma \in \Sigma$, and $s \in \Sigma^*$ if $\delta(x, \sigma)$ is defined. Note that if $\delta(x, \sigma)$ is not defined, then $\delta(x, \sigma s)$ is not defined. The generated language of $G$ is given by $L(G) = \{ s \in \Sigma^* | \delta(x_0, s) \}$, where “$\epsilon$” means “is defined.”

Definition 2: A nondeterministic finite automaton (NFA), denoted by $G_{nd}$, is a four-tuple $G_{nd} = (X, \Sigma, \delta, X_0)$, where $X$ and $\Sigma$ have the same interpretation as in a DFA, $\delta : X \times X \rightarrow 2^X$ is the (nondeterministic) state transition function, and $X_0 \subseteq X$ is the set of initial states.

By letting $B \subseteq X$ and $\sigma \in \Sigma$, $\delta(B, \sigma)$ is defined as $\cup_{x \in B} \delta(x, \sigma)$.

In order to characterize the strings generated by an NFA, the domain $X \times \Sigma$ of the transition function can be extended to $X \times \Sigma^*$. For $x \in X$, $s \in \Sigma^*$, and $\sigma \in \Sigma$, $\delta$ is defined recursively as: $\delta(x, \epsilon) = \{ x \}$; $\delta(x, \sigma s) = \delta(\delta(x, \sigma), s)$; $\delta(s, \epsilon) = \{ s \}$; $\delta(x, \epsilon) = \delta(x, \sigma s)$. An event $\sigma \in \Sigma$ is said to be feasible at state $x \in X$ if $\delta(x, \sigma)$ is nonempty. The language generated by $G_{nd}$ is defined as $L(G_{nd}) = \{ s \in \Sigma^* | \exists x \in X : \delta(x, \sigma) \neq \emptyset \}$, where $\emptyset$ denotes the empty set. The language $L(G_{nd})$ is said to be live if whenever $s \in L(G_{nd})$, there exists an event $e \in \Sigma$ such that $se \in L(G_{nd})$ [5].

The set of events $\Sigma$ in a DFA or an NFA is partitioned into the subset of observable events, $\Sigma_{obs}$, and the subset of unobservable events, $\Sigma_{uobs}$ with $\Sigma_{obs} \supseteq \Sigma \setminus \Sigma_{uobs}$. The sensor measuring unit can only observe and record observable events. The natural projection $P : \Sigma^* \rightarrow \Sigma_{obs}$ captures the sequence of observable actions in response to a sequence of events $s \in L(G_{nd})$; it is defined recursively as:

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

and $P(\sigma s) = P(\sigma) P(s)$, for $\sigma \in \Sigma, s \in \Sigma^*$. The natural projection $P$ can be used to map any trace $s \in \Sigma^*$ to the corresponding sequence of observations $P(s)$ observed at the sensor measuring unit. The inverse projection of $P$, $P^{-1} : \Sigma_{obs} \rightarrow \Sigma^*$, is defined as follows: for all $\omega \in \Sigma_{obs}$

$$P^{-1}(\omega) = \{ s \in \Sigma^* | P(s) = \omega \}.$$

A typical task by an observer/agent is to determine a set of possible states in which a system may be, or infer whether a particular (unobservable) event of interest has occurred [4]. The set of all possible states corresponding to an observable sequence $\omega \in \Sigma_{obs}$ starting from the states in $B$ is defined as $R(B, \omega) = \{ x' \in X | \exists x \in \Sigma_{obs} : \exists b \in B : [P(s) = \omega \text{ and } x' \in \delta(x, \sigma)] \}$.

The state estimation problem in DESs is defined as follows.

State Estimation Problem: [4] Given a DES described by an NFA $G_{nd}$ with a sensor measuring unit, an observer needs to determine a set of possible states based on an observation sequence $P(s) \in \Sigma_{obs}$ (generated by an underlying sequence of events $s \in L(G_{nd})$, in the given NFA) that is received from the sensor measuring unit. Following the observation of $\omega \in \Sigma_{obs}$, the state estimation problem is to compute $\hat{X}(\omega) = \{ x' \in X | \exists s' \in \Sigma_{obs} : \exists x_0 \in X_0 : P(s') = \omega \text{ and } x' \in \delta(x_0, s') \}$.

Given $\omega = P(s) \in \Sigma_{obs}$ (with $s \in L(G_{nd})$) and $\sigma_0 \in \Sigma_{obs}$, we can obtain the set $\hat{X}(\sigma_0)$ starting from the initial states in $X_0$, recursively as $\hat{X}(\epsilon) = R(X_0, \epsilon)$; $\hat{X}(\sigma_0 \epsilon) = R(\hat{X}(\omega), \sigma_0)$.

Definition 3: An observer is captured by Obs$(G_{nd}) = AC(2^X, \Sigma_{obs}, \delta_{obs}, R(X_0, \epsilon)) = (\chi_{obs}, \Sigma_{obs}, \delta_{obs}, x_{obs})$, where $2^X$ is the set of distinct subsets of $X$ (i.e., the powerset of the state set of the given NFA $G_{nd} = (X, \Sigma, \delta, X_0)$), $\Sigma_{obs}$ is the set of observable events, $x_{obs} \in 2^X$ is the set of initial states given by $x_{obs} = R(X_0, \epsilon)$, and $\delta_{obs} : 2^X \times \Sigma_{obs} \rightarrow 2^X$ is the state transition function defined for $B \in 2^X$ and $\sigma_0 \in \Sigma_{obs}$ as $\delta_{obs}(B, \sigma_0) = R(B, \sigma_0)$. AC$(\cdot)$ denotes the
III. OBSERVATION SEQUENCES UNDER ATTACKS

In general, malicious attacks may corrupt sequences at a communication channel, such that the sequence received at the sequence estimation unit is incorrect. In this section, we propose a compact way to represent possibly matching sequences and describe an efficient method to reduce the number and representation/processing of such sequences that need to be explored.

Referring to Fig. 1, if the plant generates a string \( s \in \mathcal{L}(G_{\text{sys}}) \), the observed string at the sensor measuring unit is \( \omega = \mathcal{P}(s) \). An attacker may corrupt the output signals produced by the sensor measuring unit by deleting, inserting, or substituting certain types of events. The resulting tampered observation sequence is denoted as \( \omega_A \in \mathcal{A}(\omega) \). Where \( \mathcal{A}(\omega) \) is a set of tampered sequences that can be generated by the attacker when \( \omega \) is observed. Based on \( \omega_A \), the sequence estimation unit calculates a set of matching sequences \( RA(\omega_A) \) that is used to perform state estimation.

We focus on attacks due to symbol deletions, insertions, and substitutions. In order to have a general form of attacks, suppose that each event \( \sigma_i \in \Sigma_i \) can be associated with some arbitrary replacements (for example, given \( \Sigma_i = \{\alpha, \beta, \gamma\} \), \( \alpha \) can be replaced by \( \beta \) or \( \gamma \), and some events may also be deleted or inserted under attacks. Note that we assume that each symbol (in a sequence of symbols) received at the sequence estimation unit can only be related to at most one type of attack. In other words, it is not possible for the attacker to corrupt the same observable event more than once.

More specifically, the attacker has the capability to:

1) delete certain types of events from a set \( \Sigma_D \subseteq \Sigma_i \);
2) insert certain types of events from a set \( \Sigma_I \subseteq \Sigma_i \);
3) substitute an event \( \sigma_{oi} \in \Sigma_o \) with an event \( \sigma_{oj} \in \Sigma_o \) for some pairs \( (\sigma_{oi}, \sigma_{oj}) \) of events captured in the set \( \Sigma_T \subseteq (\Sigma_o \times \Sigma_o) \setminus \{(\sigma_{oi}, \sigma_{oi})|\sigma_{oi} \in \Sigma_o\} \).

Moreover, we assume that each individual deletion, insertion, or substitution of an event is associated with a positive integer cost. Costs capture in some sense the expense of the attacker when trying to alter symbols of transmitted sequences at the communication channel.

Attack costs can be summarized by a table, as illustrated in the following example.

**Example 1:** Consider the NFA \( G_{\text{sys}} \) shown in Fig. 2, where \( X = \{0, 1, 2, 3, 4\}, \Sigma = \{\alpha, \beta, \gamma, \zeta\}, \Sigma_o = \{\alpha, \beta, \gamma\}, \Sigma_i = \{\zeta\} \) is defined in the figure, and \( X_0 = \{0, 1, 2, 3, 4\} \). Suppose that \( \Sigma_I = \{\beta\}, \Sigma_D = \{\alpha\}, \) and \( \Sigma_T = \{\{\alpha, \beta\}, \{\gamma, \alpha\}\} \).

In Table I, Column 1 represents the symbol originally generated by the system (including the empty symbol) and Row 1 shows possible corruptions due to attacks. Note that \( \epsilon \) in Column 5 means that an original event can be deleted, whereas \( \epsilon \) in Row 5 stands for insertions of events. For example, when \( \alpha \) is corrupted to \( \beta \), the attacker spends two units.

For clearer notation, we define the set of deleted labels \( D = \{d_{\sigma_{oi}}|\sigma_{oi} \in \Sigma_D\} \), where \( d_{\sigma_{oi}} \) denotes the deletion of \( \sigma_{oi} \); the set of inserted labels \( I = \{i_{\sigma_{oj}}|\sigma_{oj} \in \Sigma_I\} \), where \( i_{\sigma_{oj}} \) denotes the insertion of \( \sigma_{oj} \); and the set of substituted labels \( T = \{t_{\sigma_{oi},\sigma_{oj}}|\sigma_{oi}, \sigma_{oj} \in \Sigma_T\} \), where \( t_{\sigma_{oi},\sigma_{oj}} \) denotes the substitution of \( \sigma_{oi} \) by \( \sigma_{oj} \). The above attack forms are captured by the set of attacked labels \( AT = D \cup I \cup T \).

For example, suppose that \( \beta \) can be inserted and \( \gamma \) can be replaced by \( \epsilon \) at the communication channel under attacks. If \( \beta \) is received at the sequence estimation unit, possible original sequences could be \( \epsilon, \beta, \) or \( \gamma \). In order to clarify the type of attack, the sequences \( \epsilon, \beta, \) and \( \gamma \) are relabeled, respectively, by \( i_3, t_{3,3} \), and \( t_{3,3} \).

The corrupted projection of the attacker \( \tilde{P} : (\Sigma_o \cup AT)^* \rightarrow \Sigma_o \) captures how an attacked sequence of observations is seen at the sequence estimation unit, defined as

\[
\tilde{P}(\sigma_R) = \begin{cases} 
\sigma_R, & \text{if } \sigma_R \in \Sigma_o \\
\varepsilon, & \text{if } \sigma_R = d_{\sigma_{oi}} \in D \cup \{\varepsilon\} \\
i_{\sigma_{oj}}, & \text{if } \sigma_R = i_{\sigma_{oj}} \in I \\
t_{\sigma_{oi},\sigma_{oj}}, & \text{if } \sigma_R = t_{\sigma_{oi},\sigma_{oj}} \in T
\end{cases}
\]

and \( \tilde{P}(\omega_R) = \tilde{P}(\omega_R) \tilde{P}(\sigma_R) \) for \( \omega_R \in (\Sigma_o \cup AT)^* \), \( \sigma_R \in \Sigma_o \cup AT \cup \{\varepsilon\} \). Given \( \Omega_R \subseteq (\Sigma_o \cup AT)^* \), we also define \( \tilde{P}(\Omega_R) = \bigcup_{\omega_R \subseteq \Omega_R} \tilde{P}(\omega_R) \).

We next introduce a cost function \( \Pi_c : (\Sigma_o \cup AT)^* \rightarrow \mathbb{N} \), associating an attacked sequence with a cost, where \( \mathbb{N} = \{0, 1, 2, \ldots\} \). More specifically, \( \Pi_c \) is used to accumulate the total cost of attacks occurred at each attacked sequence. The cost function \( \Pi_c \) can be defined recursively as

\[
\Pi_c(\sigma_R) = \begin{cases} 
0, & \text{if } \sigma_R \in \Sigma_o \cup \{\varepsilon\} \\
c_{d_{\sigma_{oi}}}, & \text{if } \sigma_R = d_{\sigma_{oi}} \in D \\
c_{i_{\sigma_{oj}}}, & \text{if } \sigma_R = i_{\sigma_{oj}} \in I \\
c_{t_{\sigma_{oi},\sigma_{oj}}}, & \text{if } \sigma_R = t_{\sigma_{oi},\sigma_{oj}} \in T
\end{cases}
\]

and \( \Pi_c(\omega_R \sigma_{oi}) = \Pi_c(\omega_R) + \Pi_c(\sigma_{oi}) \), for \( \omega_R \in (\Sigma_o \cup AT)^* \), \( \sigma_{oi} \in \Sigma_o \cup AT \cup \{\varepsilon\} \). Note that we, respectively, use \( c_{d_{\sigma_{oi}}}, c_{i_{\sigma_{oj}}}, \) and \( c_{t_{\sigma_{oi},\sigma_{oj}}} \) to denote the costs of a substitution of \( \sigma_{oi} \) for \( \sigma_{oj} \), deletion

---

**Table I**

**ATTACKS WITH COSTS**

| original | \(\alpha\) | \(\beta\) | \(\gamma\) | \(\varepsilon\) |
|----------|----------|----------|----------|----------|
| \(\alpha\) | 2        | 3        |          |          |
| \(\beta\)  |          |          |          |          |
| \(\gamma\) | 1        |          |          |          |
| \(\varepsilon\) | 2        |          |          |          |
of an event $\sigma_0 \in \Sigma_D$, and insertion of an event $\sigma_0 \in \Sigma_I$, where $c_{t\alpha}^\sigma, c_{e\sigma}, c_{i\sigma}$ are positive integers.

Given a sequence of observations $\omega \in \Sigma_\omega^*$, we can systematically obtain a set of possibly tampered sequences that can be generated by the attacker. Suppose that $\omega = \sigma_0 \sigma_1 \sigma_2 \ldots \sigma_n$, where $\sigma_i \in \Sigma_\omega$ and $i \in \{1, 2, \ldots, n\}$. The set of possibly tampered sequences, denoted by $A(\omega) \subseteq \Sigma_\omega^*$, is defined as $A(\omega) = \{ A_i(\omega) \}$, where $A_i(\omega) = \{ \sigma_0 \sigma_1 \sigma_2 \ldots \sigma_{i-1} \}$.

The action projection of the attacker $P : (\Sigma_\omega \cup AT)^* \to \Sigma_\omega^*$ is defined as

$$ P(\omega R) = P(\sigma R) \quad \text{for} \quad \omega R \in (\Sigma_\omega \cup AT)^*, \quad \sigma R \in \Sigma_\omega \cup AT \cup \{ \epsilon \}. $$

Given any $\Omega R \subseteq (\Sigma_\omega \cup AT)^*$, we also define $P(\Omega R) = \cup_{\omega R \in \Omega R} P(\omega R)$.

Similarly, each sequence $\omega R \in RA(\omega)$ can be augmented with a cost value. If we set the same upper bound on the total cost to $C$ among each sequence in $RA(\omega)$, then a set of matching sequences under the maximum cost, denoted by $RA_C(\omega)$, can be obtained. It is defined as $RA_C(\omega) = \cup_{\omega R \in RA(\omega)} \{ (\omega R, c) \in \Pi L(\Omega R) | C \}$. For simplicity, we define $P(\Omega R) = \cup_{(\omega R, c) \in RA_C(\omega)} \{ (\omega R, c) \}$. $P_R(\Omega R) = \cup_{(\omega R, c) \in RA_C(\omega)} \{ (\omega R, c) \}$.

**Example 3:** Consider the string $\omega = \alpha_0 \alpha_\omega$ and $A(\omega)$ already discussed in Example 2. Assume that the attacker corruts $\omega$ to $\omega = \beta \alpha_\omega \in A(\omega)$ (the first $\alpha$ is replaced by $\beta$ consuming two units of cost. We have $RA(\omega) = \{ \beta_\omega + \beta \alpha_\omega + \beta \alpha_\omega + \beta \alpha_\omega \}$, $\sigma R \in \Sigma_\omega \cup AT \cup \{ \epsilon \}$. If we set the upper bound on the total cost to two, we can obtain $RA_2(\omega) = \{ (\beta_\omega, 0), (\beta_\omega, 1), (\beta_\omega, 2), (\beta_\omega, 3), (\beta_\omega, 4) \}$.

**IV. State Estimation Under Attacks**

In this section, we investigate tamper-tolerant state estimation problems under unconstrained and cost-constrained attacks. The goal of this section is twofold: 1) we want to formulate tamper-tolerant state estimation and to illustrate cost constraints, and 2) to obtain recursive state estimation algorithms for DESs under adversarial attacks. More specifically, by proposing appropriate transformed plants, we are able to reduce tamper-tolerant state estimation in the original plant into (standard) state estimation in the transformed plants.

A. Unconstrained Tamper-Tolerant State Estimation

State Estimation Problem Under Unconstrained Attacks: Consider a DES modeled by an NFA $G_{nd}$ and a sensor measuring unit able to measure and report the sequence of observable events $\omega = P(s')$ (with $s' \in L(G_{nd})$) to an observer. An attacker may arbitrarily intercept and alter symbols in the reported sequence of observations. The observer needs to estimate possible states according to a possibly corrupted sequence $\omega_A \in A(\omega)$ (refer to Fig. 1). Following the observation of $\omega_A$, the unconstrained tamper-tolerant state estimation problem is to compute

$$ \tilde{X}_1(\omega_A) = \{ x' \in X | \exists x_0 \in X_0, \exists x \in X^* : \omega_A \in A(P(s')) \}.$$
Fig. 3. Nondeterministic finite automaton.

The domain of δ_{tn} can be extended to X × (Σ ∪ {ε})^* in the usual way, i.e., for x ∈ X, s ∈ Σ^*, ε ∈ (Σ ∪ {ε}), we have δ_{tn}(x, es) = (∪_{s′} δ_{tn}(x, e)δ_{tn}(s′, s)).

Definition 5: Given a tampered system G_{tn}, the set of all possible states corresponding to a possibly corrupted sequence w_{A} ∈ Σ^n_{0} starting from the states in a set B with B ⊆ X is defined as R_{t}(B, w_{A}) = {x′ ∈ X | ∃s ∈ Σ^*, ∃ε ∈ B : |P(s)| = w_{A} and x′ ∈ δ_{tn}(x, s)}. Theorem 1: Given a possibly corrupted sequence w_{A} ∈ Σ^*, the tamper-tolerant state estimates \( \hat{X}_{t}(w_{A}) \) is equal to the set of state estimates in the tampered system R_{t}(X_{0}, w_{A}).

Proof: The proof is conducted by induction on the length of w_{A}. More specifically, we establish that, for all w_{A} of length |w_{A}| ∈ N, we have \( \hat{X}_{t}(w_{A}) = R_{t}(X_{0}, w_{A}) \).

1) As a base case, we consider |w_{A}| = 0 (i.e., w_{A} = ε). For any state x′ ∈ \( \hat{X}_{t}(ε) \), there exists s ∈ (Σ_{0} ∪ Σ_{1})^* such that s ∈ P(ε) and x′ ∈ δ(x, s). Suppose that s = e_{1}e_{2}...e_{n}, where e_{i} ∈ (Σ_{0} ∪ Σ_{1}) for i ∈ {1, 2, ..., n}. Let x′ = e_{0}e_{1}...e_{i−1}e_{i}.e_{i+1}...e_{n} ∈ Σ^*, where e_{i} ∈ e_{i} if e_{i} ∈ Σ_{0}, otherwise e_{i} = ε. Then, we have x′ ∈ δ_{tn}(x, s′), refer to Definition 4, which implies x′ ∈ R_{t}(X_{0}, e). We can similarly argue that for any state x′ ∈ R_{t}(X_{0}, e), we have x′ ∈ \( \hat{X}_{t}(ε) \). The base case holds.

2) Assume that the induction hypothesis holds, i.e., for all sequences w_{A} of length |w_{A}| = k, k ∈ N, we have \( \hat{X}_{t}(w_{A}) = R_{t}(X_{0}, w_{A}) \).

3) Now consider any sequence of length w_{A} of length k + 1. Clearly, w_{A} can be written as w_{A}σ_{k} for some prefix w_{A′} of length k and some observable event σ_{k} ∈ Σ_{0}. Consider any state x′ ∈ \( \hat{X}_{t}(w_{A}σ_{k}) \). This means that there exist x′, s′ ∈ (Σ_{0} ∪ Σ_{1})^* and x ∈ \( \hat{X}_{t}(w_{A}) \) such that: 1) x′ ∈ δ(x, s′σ_{k}ε) or 2) x′ ∈ δ(x, s′σ_{k}ε for some s′ ∈ Σ_{0} that satisfies (σ_{k}, A_{σ_{k}}) ∈ Σ_{1}. Then, we have x′ ∈ R_{t}((x, σ_{k}) and x ∈ R_{t}(x_{0}, w_{A}) by the induction hypothesis, which means that x′ ∈ R_{t}(X_{0}, w_{A}σ_{k}). We assume that, i.e., for any state x′ ∈ R_{t}(X_{0}, w_{A}σ_{k}), then x′ ∈ \( \hat{X}_{t}(w_{A}σ_{k}) \) can be proved similarly. This completes the proof of the induction step and the proof of the theorem. □

Given w_{A} = σ_{A_{1}}σ_{A_{2}}...σ_{A_{n}} ∈ Σ^n_{0} and σ_{A_{n+1}} ∈ Σ, we can obtain the set \( \hat{X}_{t} \) starting from the initial state in X_{0}, recursively as \( \hat{X}_{t}(ε) = R_{t}(X_{0}, ε) \), \( \hat{X}_{t}(w_{A}σ_{A_{n+1}}) = R_{t}(\hat{X}_{t}(w_{A}σ_{A_{n+1}}), σ_{A_{n+1}}) \).

The complexity of recursively computing the set of unconstrained tamper-tolerant state estimates \( \hat{X}_{t} \) is linear in the number of states and in the number of transitions, as given by |X| and |Σ|, where |X| and |Σ| are, respectively, the cardinalities of the sets X and Σ. Note that we only need to maintain the set of current states (which requires storage of O(|X|)). Moreover, observable and unobservable transitions are all considered at each state, hence the number of transitions is |Σ|. Finally, we establish that the overall complexity of each step for the recursive state estimation of \( \hat{X}_{t} \) is O(|X|).

Example 4: In the NFA in Fig. 3, Σ = {α, β, γ, ζ, σ_{f}}, Σ_{0} = {α, β, γ, ζ}, Σ_{0} = {σ_{f}}, and X_{0} = {0}. Suppose that AT = T_{x_{0}, y_{0}, z_{0}}. All transitions defined in G_{tn} are defined in G_{tn}, as shown in Fig. 4. The events generated by the attacker are partially defined in some of the states, such as γ at state 1.

Assume that s = σ_{A}αβζ is generated by G_{nd} starting from the initial state 0 such that ω = P(s) = αβζ. The attacker can corrupt ω to ω = γαζ ∈ A(ω). We have \( \hat{X}_{t}(ω_{A}) = \{3, 5\} \). For the system G_{tn}, if ω′ = ω_{A} = γαζ is observed starting from state 0, we have R_{t}({0, 5}) = \{3, 5\}.

Remark 1: The work in this section shares similarities with those in [29], [30], [32], which consider nondeterministic observations for detectability and diagnosability analysis. Nondeterministic observations can be caused by sensor faults, observation losses, or unconstrained attacks. However, the insertion of sequences of symbols that is considered in this article is not handled in those works.

B. Cost-Constrained Tamper-Tolerant State Estimation

State Estimation Problem under Cost-Constrained Attacks: Consider a DES modeled by an NFA G_{nd} and a sensor measuring unit able to measure and report the sequence of observable events ω = P(s′) [with s′ ∈ Σ(G_{nd})] to an observer. An attacker may intercept and alter, at a certain cost, the symbols in a reported sequence of observations. Given an upper bound C on the total cost that the attacker can utilize, the observer needs to estimate possible states (and their associated costs) starting from an initial state associated with zero cost in X_{0} (with X_{0} = {(x_{0}, 0)|x_{0} ∈ X_{0}}), according to a possibly corrupted sequence w_{A} with (w_{A}, ε) ∈ A_{C}(ω) received from the sequence estimation unit (refer to Fig. 1). The tamper-tolerant state estimation problem under cost constraints is to compute \( \hat{X}_{t}(ω_{A}) = \{x′, ε \} ∈ X × \{0, 1, ..., C\} | \exists(x_{0}, 0) ∈ X_{0}, 3s \in Y : (w_{A}, ε) ∈ A_{C}(P(s)) and x′ ∈ \hat{X}_{t}(ω_{A}) \}.

Note that it is possible that \( \hat{X}_{t}(ω_{A}) \) contains the same state associated with different costs. As in the unconstrained case, we can construct a modified system G_{nmd}(C), which contains all possible corruptions of an attacker under the maximum cost C. With the modified system, tamper-tolerant state estimation in G_{nd} under cost-constrained attacks is transformed into (standard) state estimation in G_{nmd}(C) (with costs incorporated in the states).

Definition 6: Consider an NFA G_{nd} = (X, Σ, δ, X_{0}) generating observations that can be tampered via a set AT = D ∪ I ∪ T of deletions, insertions, and substitutions, under a maximum cost C. A modified system, denoted by G_{nmd}(C), is a four-tuple NFA: G_{nmd}(C) = (X_{mn}, Σ ∪ {ε}, δ_{mn}, X_{mn0}), where X_{mn} ⊆ X × \{0, 1, 2, ..., C\} is the set of states, each associated with its respective cost; Σ is the set of events; the set of initial states X_{mn0} = {(x, 0)|x ∈ X_{0}} ⊆ X_{mn} is associated with zero initial cost. The state transition function δ_{mn} : X_{mn} × Σ ∪ {ε} → 2^{X_{mn}} is defined as follows: for (x, ε) ∈ X_{mn}, ε ∈ Σ ∪ {ε}, σ_{oi} ∈ Σ_{oi}, δ_{mn}(x, c, ε) = N_{0} ∪ N_{T} ∪ N_{D} ∪ N_{I} with 1) the zero cost set N_{0} = \{0 \} × \{x, ε \} × \{c \} if c ∈ Σ; 2) the deletion set N_{D} = \{ (x, σ_{oi}) × \{e \} if (d_{ei} ∈ D) ∧ (e = ε) ∧ (c_{dei} + c ≤ C),(c_{dei} + c < C), otherwise.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
3) the insertion set
\[ N_I = \begin{cases} \{x\} \times \{c_i + c\}, & \text{if } (i_x \in I) \land (c_i + c \leq C), \\ \emptyset, & \text{otherwise} \end{cases} \]

4) the substitution set
\[ N_F = \begin{cases} \delta(x, \sigma_e) \times \{c_{x_e + c}\}, & \text{if } (t_{\sigma_e} \in T) \land (c_{x_e + c} \leq C), \\ \emptyset, & \text{otherwise} \end{cases} \]

The domain of \( \delta_{mn} \) can be extended to \( X_{mn} \times \Sigma^* \) in the usual way, i.e., for \( x_{mn} \in X_{mn}, \sigma \in \Sigma \), we have \( \delta_{mn}(x_{mn}, \sigma) = \bigcup x_{mn, m, s} \delta_{ms}(x_{mn}, \sigma) \delta_{sn}(x_{sn}, s) \).

**Definition 7:** Given a modified system \( G_{\text{mnd}}(C) \), the set of pairs of possible states associated with costs corresponding to a possibly corrupted sequence \( \omega_A \) with \( (\omega_A, c) \in A_C(\omega) \) starting from the states associated with costs in a set of pairs \( B_C \subseteq X \times \{0, 1, \ldots, C\} \) is defined as
\[ R_{ct}(B_C, \omega_A) = \{(x', c') \in X \times \{0, 1, \ldots, C\} | \exists s \in \Sigma^* \} \exists \hat{s} \in \Sigma^*, \exists (x, c') \in B_C : [P(s) = \omega_A \land (x', c') \in \delta_{mn}(x, c')] \].

The proof of the following theorem is similar to the proof of Theorem 1 and is omitted.

**Theorem 2:** Given a possibly corrupted sequence \( \omega_A \in \Sigma_n \), the tamper-tolerant state estimates under cost constraints \( \hat{X}_{ct}(\omega_A) \) is equal to the set of state estimates in the modified system \( R_{ct}(X_{ct0}, \omega_A) \).

Given \( \omega_A = \sigma_{A1}\sigma_{A2} \ldots \sigma_{An} \in \Sigma^*_n \) and \( \sigma_{An+1} = \in \Sigma_n \), we can obtain the set \( \hat{X}_{ct} \) starting from initial states with zero cost in \( X_{ct0} \), recursively as \( \hat{X}_{ct}(\varepsilon) = R_{ct}(X_{ct0}, \varepsilon) \); \( \hat{X}_{ct}(\omega_A \sigma_{An+1}) = R_{ct}(\hat{X}_{ct}(\omega_A), \sigma_{An+1}) \).

The complexity of recursively conducting the set of cost-constrained tamper-tolerant state estimates \( \hat{X}_{ct} \) is linear in the number of states associated with costs and in the number of transitions given by \(|X|(|C + 1) + |\Sigma|\), respectively. We can establish that the overall complexity of each step for the recursive state estimation of \( \hat{X}_{ct} \) is \( O(|X||\Sigma|C) \).

**Example 5:** Continuing with Example 4, suppose that the total cost of \( c_{t0} = 1 \), \( c_{t1} = 1 \), and the upper bound on the total cost \( C = 1 \). If the attacker corrupts \( \omega = \alpha \beta \gamma \) to \( \omega' = \gamma \beta \gamma \) consuming one unit of cost, we have \( \hat{X}_{ct}(\omega') = \{(3, 1)\} \). If the upper bound on the total cost is set to \( C = 2 \) and the attacker corrupts \( \omega \) to \( \omega'' = \gamma \alpha \gamma \), then we have \( \hat{X}_{ct}(\omega'') = \{(3, 2), (5, 0)\} \).

We construct \( G_{\text{mnd}}(1) \) and \( G_{\text{mnd}}(2) \) to perform cost-constrained tamper-tolerant state estimation for \( G_{\text{nd}} \) under the upper bounds \( C = 1 \) and \( C = 2 \), respectively. The modified NFA in Definition 6 is shown in Fig. 5. For example, \( G_{\text{mnd}}(1) \) reaches state (2,1) from state (1,0) if the attacker corrupts \( \alpha \) to \( \gamma \) consuming one unit of cost. Once \( G_{\text{mnd}}(1) \) is constructed, we can perform state estimation for \( \gamma \beta \gamma \) in the standard way and obtain one pair of a possible state and its cost as \( \{(3, 1)\} \). Compared with Example 4, the attacker corrupts \( \omega \) to \( \omega' \) rather than \( \omega_A \) due to the upper bound on the total cost.

### C. Least Cost Tamper-Tolerant State Estimation

In this section, we aim to obtain the least cost tamper-tolerant state estimates using a standard Viterbi-like algorithm via a corrupted system with costs on the various transitions [33]. This method can find the least-cost path(s) from each initial state to each reachable state according to a possibly corrupted sequence \( \omega_A \in \Sigma_n^* \).

**Definition 8:** Given an NFA \( G_{\text{nd}} = (X, \Sigma, \delta, X_0) \), a corrupted system, denoted by \( G_{\text{cn}} \), is an NFA \( G_{\text{cn}} = (X, \Sigma \cup \{\varepsilon\}) \times N, \delta_{\text{cn}}, X_0) \), where \( (\Sigma \cup \{\varepsilon\}) \times N \) is the set of pairs involving an event and its corresponding cost. The state transition function \( \delta_{\text{cn}} : X \times (\Sigma \cup \{\varepsilon\}) \times N \rightarrow 2^X \) is defined as follows: for \( x \in X, (e, c) \in (\Sigma \cup \{\varepsilon\}) \times N, \sigma_{oi} \in \Sigma_n \),

\[ \delta_{cn}(x, (e, c)) = \begin{cases} \delta(x, e), & \text{if } c = 0, \\ \delta(x, \sigma_{oi}), & \text{if } (e, c) \in (\Sigma_{oi} \times N) \land (c = c_{\sigma_{oi}} > 0), \\ \{x\}, & \text{if } (e, c) \in (\Sigma \times N) \land (c = c_{\sigma_{oi}} > 0), \\ \delta(x, \sigma_{oi}), & \text{if } (\sigma_{oi}, e) \in \Sigma \land (c = c_{\sigma_{oi}} > 0), 
\end{cases} \]

The domain of \( \delta_{cn} \) can be extended to \( X \times (\Sigma \cup \{\varepsilon\}) \times N \) in the usual way, i.e., for \( x \in X, \sigma_{ei} \in (\Sigma \cup \{\varepsilon\}) \times N \), we have \( \delta_{cn}(x, (e, c)) = \cup x_{ei, e, c} \delta_{ei}(x, (e, c)) \). All transitions defined in \( G_{\text{nd}} \) are set with zero cost in \( G_{\text{cn}} \), as shown in Fig. 6. The pairs involving events and positive costs are also partially defined in some of the states, such as \( (\gamma, 1) \) at state 1.

**Definition 9:** Given a set of pairs of possible states associated with costs \( B_C \subseteq X \times \{0, 1, \ldots, C\} \), the set of all possible states associated with costs corresponding to a possibly corrupted sequence \( \omega_A \in \Sigma_n^* \) is defined as
\[ R_{ct}(B_C, \omega_A) = \{(x', c') \in X \times \{0, 1, \ldots, C\} | \exists s_e \in (\Sigma \cup \{\varepsilon\}) \times N : [P(s) = \omega_A \land (x', c') \in \delta_{cn}(x, s_e)] \}
\]

where \( s_e = (e_1, c_1)(e_2, c_2) \ldots (e_n, c_n) \) and \( s = e_1e_2 \ldots e_n \in \Sigma^* \).
Since we are interested in possible states associated with the minimal costs, a possible state and its cost \((x', c') \in R_c(B_C, \omega_A)\) is retained if there does not exist \((x', c') \in R_c(B_C, \omega_A)\) such that \(c < c'\), which is formally defined as \(R_{c, \text{min}}(B_C, \omega_A) = \{(x', c') \in X \times \{0, 1, \ldots, C\} | \exists (x', c') \in R_c(B_C, \omega_A), \tilde{c}(x', c') < c\} = c < c'\).

Given \(\omega_A = \sigma_{A_1} \sigma_{A_2} \ldots \sigma_{A_m} \in \Sigma^m\) and \(\sigma_{A_{m+1}} \in \Sigma_m\), we can define the least cost state estimation of \(\hat{x}_{\text{min}}\) starting from initial states with zero cost in \(X_{CD}\), recursively as \(\hat{x}_{\text{min}}(\epsilon) = R_{c, \text{min}}(X_{CD}, \epsilon); \hat{x}_{\text{min}}(\omega_{A_{m+1}}) = R_{c, \text{min}}(\hat{x}_{\text{min}}(\omega_A), \sigma_{A_{m+1}})\).

The complexity of performing the least cost tamper-tolerant state estimation \(\hat{x}_{\text{min}}\) depends on the number of states associated with minimal costs and on the number of transitions given by \(|X|\) and \(|\Sigma|\), respectively. We can establish that the overall complexity of each step for the state estimation of \(\hat{x}_{\text{min}}\) is \(O(|X|^2|\Sigma|)\). Note that in \(\hat{x}_{\text{min}}\), we need to retain both a reachable state and its previous state at each step (which requires storage of \(O(|X|^2\))

Proposition 1: Given a possibly tampered sequence \(\omega_A = \sigma_{A_1} \sigma_{A_2} \ldots \sigma_{A_m} \in \Sigma^m\), any state with the least cost \((x, c) \in R_c(X_{CD}, \omega_A)\) is in \(\hat{x}_{\text{min}}(\omega_A)\).

Proof: By contradiction, suppose that there is a state \((x', c') \in R_c(X_{CD}, \omega_A)\) and \((x', c') \notin \hat{x}_{\text{min}}(\omega_A)\). Let \((x_i, c_i) \in R_c(X_{CD}, \omega_A)\) and \((x_i, c_i) \notin R_c((x_i, c_i), \omega_A)\), where \(\omega_A = \sigma_{A_1} \sigma_{A_2} \ldots \sigma_{A_j} \omega_{A_1} = \sigma_{A_{j+1}} \sigma_{A_{j+2}} \ldots \sigma_{A_m}\) for \(j \in \{0, 1, \ldots, m\}\). Suppose that \((x_i, c_i)\) is deleted while calculating \(\hat{x}_{\text{min}}(\omega_A)\). This means that there exists \((x_i, c_i) \in \hat{x}_{\text{min}}(\omega_A)\) such that \(c_i < c\). Let the total cost of \(\omega_{A_1}\) be \(c_1\). If \(R_c((x_i, c_i), \omega_{A_1}) \neq 0\), \(R_c((x_i, c_i), \omega_{A_1})\) must be nonempty and can lead to the same state \(x'\) of the plant. Since \(c_1 + c_2 + c_3 + c_4 = c'\), we have \((x', c') \in R_c(X_{CD}, \omega_A)\), which contradicts that \((x', c')\) is the estimated state with the least cost.

Example 7: Consider again the system in Fig. 6. The process of the least cost state estimation corresponding to \(\omega_A = \gamma\alpha\) is shown in Fig. 7. It starts at states 0 and 1 associated with zero cost. If an event \(\gamma\) is observed, \(\hat{x}_{\text{min}}(\gamma) = \{(2, 1), (4, 0)\}\) holds. If \(\alpha\) is subsequently observed after \(\gamma\) under the upper bound on the total cost \(C = 3\), we have \(\hat{x}_{\text{min}}(\gamma\alpha) = R_{c, \text{min}}(\hat{x}_{\text{min}}(\gamma), \alpha\gamma) = \{(3, 2), (5, 0)\}\). Note that (3, 3) does not appear in \(\hat{x}_{\text{min}}(\gamma\alpha)\) (marked with a dotted line) since only state 3 associated with the minimal cost two is retained at this step.

V. Conclusion

In this article, we consider current-state estimation in a DES modeled as an NFA, under insertions, deletions, and substitutions of observed symbols. All possibly matching sequences of observations are represented in a compact way, which avoids explicitly enumerating all such sequences. In order to perform tamper-tolerant state estimation, different transformed systems are constructed, where attacks and costs are attached to the original plant. Then, we argue that we can perform recursive tamper-tolerant state estimation for the given plant under attacks through (standard) state estimation in the transformed plants. A recursive least cost tamper-tolerant state estimation method is proposed with complexity of \(O(|X|^2|\Sigma|)\).

In the future, we plan to develop ways to efficiently assess whether it is preferable to perform state estimation under multiple sensor measuring units. We also plan to consider how state estimation can be achieved in the presence of other types of attacks or cost assignment functions.

References

[1] D. Simon, Optimal State Estimation: Kalman, H. Infinity, and Nonlinear Approaches. Hoboken, NJ, USA: Wiley, 2006.
[2] A. Monticelli, State Estimation in Electric Power Systems: A Generalized Approach. Kluwer, Berlin, Germany: Springer, 2012.
[3] P. J. Ramadge and W. M. Wonham, “Supervisory control of a class of discrete event processes,” SIAM J. Control Optim., vol. 25, no. 1, pp. 206–230, 1987.
[4] C. Hadjicostis, Estimation and Inference in Discrete Event Systems. Berlin, Germany: Springer, 2020.
[5] M. Sampaio, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, “Diagnosability of discrete-event systems,” IEEE Trans. Autom. Control, vol. 40, no. 9, pp. 1555–1575, Sep. 1995.
[6] R. Debouk, S. Lafortune, and D. Teneketzis, “Coordinated decentralized protocols for failure diagnosis of discrete event systems,” Discrete Event Dynamic Syst., vol. 10, no. 1–2, pp. 33–86, 2000.
[7] R. Debouk, S. Lafortune, and D. Teneketzis, “On the effect of communication delays in failure diagnosis of decentralized discrete event systems,” Discrete Event Dynamic Syst., vol. 13, no. 3, pp. 263–289, 2003.
[8] S. L. Shu, F. Lin, H. Ying, and X. G. Chen, “State estimation and detectability of probabilistic discrete-event systems,” Automatica, vol. 44, no. 12, pp. 3054–3060, 2008.
[9] S. L. Shu, F. Lin, and H. Ying, “Detectability of discrete event systems,” IEEE Trans. Autom. Control, vol. 52, no. 12, pp. 2356–2359, Dec. 2007.
[10] X. Yin, Z. J. Li, and W. L. Wang, “Trajectory detectability of discrete-event systems,” Syst. Control Lett., vol. 119, pp. 101–107, 2018.
[11] J. W. Bryans, M. Koutny, L. Mazaré, and P. Y. Ryan, “Opacity generalised to transition systems,” Int. J. Inf. Secur., vol. 7, no. 6, pp. 421–435, 2008.
[12] A. Saboori and C. N. Hadjicostis, “Notions of security and opacity in discrete event systems,” in Proc. Conf. Decis. Control, 2007, pp. 5056–5061.
[13] A. Saboori and C. N. Hadjicostis, “Verification of initial-state opacity in security applications of discrete event systems,” Inf. Sci., vol. 246, pp. 115–132, 2013.
[14] A. Saboori and C. N. Hadjicostis, “Opacity-enforcing supervisory strategies for secure discrete event systems,” in Proc. Conf. Decis. Control, 2008, pp. 889–894.
[15] M. Wakaiki, P. Tabuada, and J. P. Hespanha, “Supervisory control of discrete-event systems under attacks,” Dyn. Games Appl., pp. 965–983, 2019.
[16] L. Hu, Z. D. Wang, Q. L. Han, and X. H. Liu, “State estimation under false data injection attacks: Security analysis and system protection,” Automatica, vol. 87, pp. 176–183, 2018.
[17] E. Athanasopoulos and C. N. Hadjicostis, “Maximum likelihood failure diagnosis in finite state machines under unreliable observations,” Automatica, vol. 55, no. 3, pp. 579–593, Mar. 2010.
[18] L. K. Carvalho, M. V. Moreira, and J. C. Basilio, “Generalized robust diagnosability of discrete event systems,” in Proc. IFAC World Congr., 2011, pp. 8737–8742.
[19] R. M. Góes, E. Kang, R. Kwong, and S. Lafortune, “Stealthy deception attacks for cyber-physical systems,” in Proc. Conf. Decis. Control, 2017, pp. 3200–3205.
[20] R. Su, “A cyber attack model with bounded sensor reading alterations,” in Proc. Amer. Control Conf., 2017, pp. 3200–3205.
[21] D. Thorsley, T. S. Yoo, and H. E. Garcia, “Diagnosability of stochastic discrete-event systems under unreliable observations,” in Proc. Amer. Control Conf., 2008, pp. 1118–1165.
[22] R. K. Boel and J. H. Van Schuppen, “Decentralized failure diagnosis for discrete-event systems with costly communication between diagnosers,” in Proc. Int. Workshop Discrete Event Syst., 2002, pp. 175–181.
[24] M. Khanna, “Sampling and transmission policies for controlled markov processes with costly communication,” Ph.D. thesis, Dept. Elect. Eng., Univ. Toronto, Toronto, 1973.

[25] F. Lin, W. Wang, L. T. Han, and B. Shen, “State estimation of multi-channel networked discrete event systems,” IEEE Trans. Control Netw. Syst., vol. 7, no. 1, pp. 53–63, Mar. 2020.

[26] D. Ding, Q. L. Han, Y. Xiang, X. H. Ge, and X. M. Zhang, “A survey on security control and attack detection for industrial cyber-physical systems,” Neurocomputing, vol. 275, pp. 1674–1683, 2018.

[27] E. Mousavinejad, F. W. Yang, Q. L. Han, and L. Vlacic, “A novel cyber attack detection method in networked control systems,” IEEE Trans. Cybern., vol. 48, no. 11, pp. 3254–3264, Nov. 2018.

[28] A. Householder, A. Manion, L. Pesante, G. M. Weaver, and R. Thomas, “Managing the threat of denial-of-service attacks,” Carnegie-Mellon Univ. Software Engineering Inst., Pittsburgh, PA, USA, Report, 2001.

[29] S. Takai and T. Ushio, “Decentralized diagnosis of discrete event systems modeled by mealy automata with nondeterministic output functions,” in Proc. Amer. Control Conf., 2010, pp. 2613–2618.

[30] S. Takai and T. Ushio, “Verification of codiagnosability for discrete event systems modeled by mealy automata with nondeterministic output functions,” IEEE Trans. Autom. Control, vol. 57, no. 3, pp. 798–804, Mar. 2012.

[31] C. G. Cassandras and S. Lafortune, Introduction to Discrete Event Systems. Berlin, Germany: Springer, 2009.

[32] L. Zhou, S. L. Shu, and F. Lin, “Detectability of discrete-event systems under nondeterministic observations,” IEEE Trans. Automat. Sci. Eng., vol. 18, no. 3, pp. 1315–1327, Jul. 2021.

[33] G. D. Forney, “The viterbi algorithm,” in Proc. IEEE, 1973, vol. 61, no. 3, pp. 268–278.