Title

Securing state reconstruction under sensor and actuator attacks: Theory and design

Authors

Showkatbakhsh, Mehrdad

Shoukry, Yasser

Diggavi, Suhas N

et al.

Publication Date

2020-06-01

DOI

10.1016/j.automatica.2020.108920

Securing State Estimation Under Sensor and Actuator Attacks: Theory and Design

Mehrdad Showkatbakhsh [a], Yasser Shoukry [b], Suhas Diggavi [a], Paulo Tabuada [a]

[a] *Electrical & Computer Engineering Department, UCLA, Los Angeles, CA*

[b] *Electrical & Computer Engineering, University of Maryland, College Park, MD*

Abstract

This paper discusses the problem of estimating the state of a linear time invariant system when some of its sensors and actuators are compromised by an adversarial agent. In the model considered in this paper, the malicious agent attacks an input (output) by manipulating its value arbitrarily, i.e., we impose no constraints (statistical or otherwise) on how control commands (sensor measurements) are changed by the adversary. In the first part of this paper, we introduce the notion of sparse strong observability and we show that is a necessary and sufficient condition for correctly reconstructing the state despite the considered attacks. In the second half of this work, we propose an estimator to harness the complexity of this intrinsically combinatorial problem, by leveraging satisfiability modulo theory solving. Numerical simulations demonstrate the effectiveness and scalability of our estimator.

Key words: Cyber-Physical Security, State Estimation, Security Monitoring

1 Introduction

Cyber-Physical Systems (CPS) are characterized by the tight interconnection of cyber and physical components. CPS are not only prone to actuator and sensor failures but also to adversarial attacks on the control and sensing modules. Security of CPS is no longer restricted to the cyber domain, and recent incidents such as the StuxNet malware [20] and the security flaws reported on modern cars [13, 19] motivated the recent interest in security of CPS, (see for example, [1, 6, 26, 41] and references therein). During the last decade, a number of security problems have been tackled by the control community, *e.g.,* denial-of-service [8, 14, 34, 45], replay attacks [27], man-in-the-middle attacks [40], false data injection [25], etc.

This paper addresses the problem of state estimation when several sensors *and* actuators are under attack. We broadly refer to state estimation in the adversarial environment as secure state estimation. Our attack model is quite general and we impose no constraints on the magnitude, statistical properties, or temporal characteristics of the signals manipulated by the adversary.

Secure state estimation has gained the attention of the control community over the past decade [12]. In one line of work, the problem of state estimation and control under sensor attacks is investigated and the authors derived necessary and sufficient conditions under which estimation and stabilization are possible [11]. Shoukry et. al. [36] further refined this condition and called it sparse observability. Chong et. al. [7] found an equivalent condition for continuous-time systems and called it observability under attack. Nakahira et. a. [29] investigated a similar problem while considering the asymptotic correctness of state estimation. The authors relaxed the sparse observability condition to sparse detectability and showed it is a necessary and sufficient condition for asymptotic correctness. The noisy version of this problem has been investigated in the literature [2,3,23,24,28]. Mishra et. al. [23] derived the optimal solution for Gaussian noise. In this paper, we solve the more general problem of *actuator and sensor* attacks that includes, as a special case, sensors attacks.

Under the sparse attack model in which an adversary can only target a bounded number of actuators and sensors, state estimation is intrinsically a combinatorial problem. Shoukry et. al. [35] proposed a novel secure state estimator using the Satisfiability Modulo Theory (SMT) paradigm, called IMHOTEP-SMT. The authors only considered attacks on sensors. In this paper we address the more general problem of sensor *and* actuator attacks and build an SMT-based es-

⋆ This paper was not presented at any IFAC meeting. Corresponding author M. Showkatbakhsh Tel. +213 364-8655.

Email addresses: mehrdadsh@ucla.edu (Mehrdad Showkatbakhsh), yshoukry@ece.umd.edu (Yasser Shoukry), suhas@ee.ucla.edu (Suhas Diggavi), tabuada@ucla.edu (Paulo Tabuada).

timator that can correctly reconstruct the state under both types of attacks.

In another line of work, the problem of secure state estimation has been studied when the exact model of the system is not available [30, 43]. Tiwari et. al. [42] proposed an online learning method by building so-called safety envelopes as it receives attack-free data to detect abnormality in the data when the system is prone to attacks. In [38, 39] the authors considered system identification under sensors attacks. In all of these works, the adversarial agent is restricted to only attacking sensors.

Pasqualetti et. al. [31] investigated the problem of attack detection and identification. The authors related the undetectable and unidentifiable attacks to the zero-dynamics of the underlying system. The proposed attack identification mechanism consists of a number of fault-monitor filters that provide formal guarantees for the existence of the attack. The number of filters, however, grows exponentially with the number of attacked sensors/actuators, and therefore hinders scalability. In another work [33], the authors investigated detectibility and identifiability of attacks in the presence of disturbances and the concept of security index is generalized to dynamical systems. The proposed method is inherently combinatorial and does not scale well with the number of attacked sensors and actuators. In this paper, by leveraging the SMT paradigm, we design a state estimator that scales well with the number of sensors and actuators.

Fault isolation and fault detection filters are classical control topics closely related to secure state estimation. The traditional fault tolerant filters can detect faults on actuators and sensors, however, they are not adequate for the purpose of security. Some of these filters assume a priori knowledge (statistical or temporal) of the fault signals [5], an assumption that does not hold in the security framework. The classical fault detection filters [17] do not guarantee identification of all possible adversarial signals and zero-dynamics attacks remain stealthy. As an alternative approach, robustification has been used in order to estimate the state despite sparse attacks by either deploying Kalman filters or principle component analysis [10, 22]. The main drawback of these methods is the absence of formal guarantees for the correctness of the state. In contrast, the method proposed in this paper is guaranteed to construct the state correctly in spite of attacks on sensors *and/or* actuators if the number of attacked components is below a specified threshold that depends on the system. In a recent work [15], Harirchi et. al. proposed a novel fault detection approach using techniques from model invalidation. The authors pursued a worst-case scenario approach and therefore their framework is suitable for security. However, necessary and sufficient conditions for state estimation in a general adversarial setting were not investigated in [15]. In this paper, we precisely characterize the class of systems, by providing necessary and sufficient conditions, for which state reconstruction is possible despite sensor and/or actuator attacks.

The contributions of this paper can be summarized as follows:

- We introduce the notion of sparse strong observability by drawing inspiration from sparse observability [11,36] and the classical notion of strong observablity [16]. We show this is the relevant property when the adversarial agent not only compromises sensor measurements but can also attack inputs.
- We develop an observer by leveraging the SMT approach to harness the exponential complexity of the problem. Our observer consists of two blocks interacting iteratively until the true state is found (see Section 4 for a detailed explanation of the observer's architecture).
- We propose two methods to further decrease the running time of the proposed algorithm by reducing the number of iterations of the observer. The first method exploits heuristics that can be efficiently computed at each iteration. The second method is inspired by the QUICKXPLAIN algorithm [18] that efficiently finds an irreducibly inconsistent set (see Section 4 for a detailed discussion on the aforementioned methods). We demonstrate the scalability of our proposed observer by several numerical simulations.

A preliminary version of some of the results in this paper were presented in [37] where we introduced the notion of sparse strong observability and drew the connection to secure state estimation. However, the formal proofs were not provided due to space limitations. Furthermore, we propose a new observer that outperforms the observer introduced in [37]. This paper is organized as follows. Section 2 introduces notation followed by the attack model and the precise problem formulation. In Section 3, we introduce the notion of sparse strong observability and relate this notion to the problem of state reconstruction when some of the inputs and outputs are under adversarial attacks. This section concludes with the main theoretical contribution of this paper that is Theorem 8. Section 4 is devoted to designing an observer by exploiting the SMT paradigm. Section 5 provides the simulation results followed by Section 6 that concludes the paper.

2 Problem Definition

2.1 Notation

We denote the sets of real, natural and binary numbers by $\mathbb{R}$, $\mathbb{N}$ and $\mathbb{B}$. We represent vectors and real numbers by lowercase letters, such as u, x, y, and matrices with capital letters, such as A. Given a vector $x \in \mathbb{R}^n$ and a set $O \subseteq \{1,\dots,n\}$, we use $x|_O$ to denote the vector obtained from x by removing all elements except those indexed by the set O. Similarly, for a matrix $C \in \mathbb{R}^{n_1 \times n_2}$ we use $C|_{(O_1,O_2)}$ to denote the matrix obtained from C by eliminating all rows and columns except the ones indexed by O_1 and O_2, respectively, where $O_i \subseteq \{1,\dots,n_i\}$ with $n_i \in \mathbb{N}$ for $i \in \{1,2\}$. In order to simplify the notation, we use $C|_{(.,O_2)} := C|_{(\{1,\dots,n_1\},O_2)}$ and $C|_{(O_1,.)} := C|_{(O_1,\{1,\dots,n_2\})}$. We denote the complement of O by $\overline{O} := \{1,\dots,n\} \setminus O$. We use the notation $\{x(t)\}_{t=0}^{T-1}$ to denote the sequence $x(0),\dots,x(T-1)$, and we drop the sub(super)scripts whenever it is clear from the context.

A Linear Time Invariant (LTI) system is described by the following equations:

$$\begin{aligned} x(t+1) &= Ax(t)+Bu(t),\\ y(t) &= Cx(t)+Du(t), \end{aligned} \tag{1}$$

where $u(t)\in\mathbb{R}^m$, $x(t)\in\mathbb{R}^n$ and $y(t)\in\mathbb{R}^p$ are the input, state and output variables, respectively, $t\in\mathbb{N}\cup\{0\}$ denotes time, and A, B, C and D are system matrices with appropriate dimensions. We use (A,B,C,D) to denote the system described by (1). The order of an LTI system is defined as the dimension of its state space. A trajectory of the system consists of an input sequence with its corresponding output sequence. For an LTI system,

$$\mathscr{O}_{(A,C)} := \begin{bmatrix} C^T & A^TC^T & \dots & (A^T)^{n-1}C^T \end{bmatrix}^T, \tag{2}$$

$$\mathscr{N}_{(A,B,C,D)} := \begin{bmatrix} D & 0 & \dots & 0\\ CB & D & \dots & 0\\ \vdots & & \ddots & \\ CA^{n-2}B & CA^{n-3}B & \dots & D \end{bmatrix}, \tag{3}$$

are the *observability* and *invertibility* matrices, respectively, where n is the order of the underlying system. In this paper, we often work with subsets of inputs and outputs. For a subset of outputs $\Gamma_y\subseteq\{1,\dots,p\}$, we use the notation $\mathscr{O}_{\Gamma_y} := \mathscr{O}_{(A,C|_{(\Gamma_y,.)})}$ to denote the observability matrix of outputs in the set Γ_y. For a set of inputs $\Gamma_u\subseteq\{1,\dots,m\}$, we use the notation $\mathscr{N}_{\Gamma_u\to\Gamma_y}$ to denote $\mathscr{N}_{(A,B_{(.,\Gamma_u)},C_{(\Gamma_y,.)},D_{(\Gamma_y,\Gamma_u)})}$. For $x\in\mathbb{R}^n$, we define its support set as the set of indices of its non-zero components, denoted by $\mathrm{supp}(x)$. Similarity we define the support of the sequence $\{x(t)\}$ as $\mathrm{supp}(\{x(t)\}) := \cap_t \mathrm{supp}(x(t))$. The observer proposed in this paper uses batches of inputs and outputs in order to reconstruct the state. We reserve capital bold letters to denote these batches,

$$\mathbf{Y}^\tau(t) := \begin{bmatrix} y(t-\tau+1)^T & \dots & y(t)^T \end{bmatrix}^T, \tag{4}$$

$$\mathbf{U}^\tau(t) := \begin{bmatrix} u(t-\tau+1)^T & \dots & u(t)^T \end{bmatrix}^T, \tag{5}$$

where $\tau\le n$. Whenever τ is the order of the underlying system, we may drop the superscript for ease of notation. For a subset of outputs (inputs), denoted by $\Gamma_y\subseteq\{1,\dots,p\}$ ($\Gamma_u\subseteq\{1,\dots,m\}$), we use the notation $\mathbf{Y}^\tau|_{\Gamma_y}(t)$ ($\mathbf{U}^\tau|_{\Gamma_u}(t)$) for the batches of length τ that only consists of outputs (inputs) in the set Γ_y (Γ_u). For a vector $x\in\mathbb{R}^n$, we denote a generic norm, l_2-norm and l_1-norm of x by $\|x\|$, $\|x\|_2$ and $\|x\|_1$.

2.2 System and Attack model

This work is concerned with the problem of state reconstruction of LTI systems. We consider the scenario in which sensors and actuators are both prone to adversarial attacks. The ultimate goal is to reconstruct the state despite these attacks. In this part, we define the attack model and conclude this section with the precise problem statement. The system S, is described by the following equations:

$$\begin{aligned} x(t+1) &= Ax(t)+Bu_S(t),\\ y_S(t) &= Cx(t)+Du_S(t). \end{aligned} \tag{6}$$

Without loss of generality we assume $\begin{bmatrix} B^T & D^T \end{bmatrix}^T$ to be of full column rank.

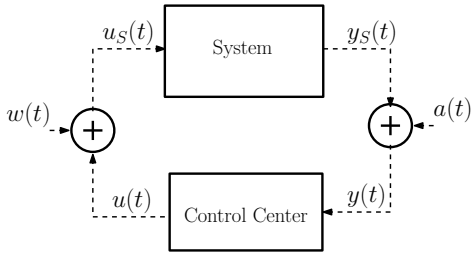

Fig. 1. The generic attack model considered in this paper.

Each actuator (sensor) corresponds to one input (output) and we use input (output) instead of actuator (sensor) in the rest of this paper. In this set up the adversary can attack both inputs and outputs. We model these attacks by additive terms and by imposing a sparsity constraint on them,

$$\begin{cases} u_S(t) &= u(t)+w(t),\\ y(t) &= y_S(t)+a(t), \end{cases} \tag{7}$$

where $u(t)\in\mathbb{R}^m$ and $y(t)\in\mathbb{R}^p$ are the controller-designed input and the observed output, respectively, and $w(t)\in\mathbb{R}^m$ and $a(t)\in\mathbb{R}^p$ are signals injected by the malicious agent. In the rest of this paper, we refer to these signals $(w(t),a(t))$ as the attack of the adversarial agent. We use the subscript S for signals that directly come from/to the system. The controller can only observe $y(t)$ and compute the input $u(t)$. This generic attack model is depicted in Figure 1.
When the adversary attacks an input (output) it can change its value to any arbitrary number without explicitly revealing its presence. The only limitation that we impose on the power of the malicious agent is the maximal number of inputs and outputs that can be attacked.

Assumption 1 (Bound on the number of attacks) *The number of inputs and outputs under attack are bounded by r and s, respectively.*

Therefore, the malicious agent can attack a subset of inputs and outputs denoted by $\Gamma_u\subseteq\{1,\dots,m\}$ and $\overline{\Gamma}_y\subseteq\{1,\dots,p\}$,[1] respectively, with $|\Gamma_u|\le r$ and $|\overline{\Gamma}_y|\le s$,

[1] For ease of exposition, we use Γ_u to denote under-attack inputs while using Γ_y for the set of attack-free outputs, i.e., the set of under-attack outputs is represented by $\overline{\Gamma}_y := |\{1,\dots,p\}\setminus\Gamma_y$ in this paper.

such that $\text{supp}(\{w(t)\}) \subseteq \Gamma_u$ and $\text{supp}(\{a(t)\}) \subseteq \overline{\Gamma}_y$. Note that these sets are not known to the controller and only upper bounds on their cardinality are given. Once the adversary chooses these sets, inputs and outputs outside these sets remain attack-free. This assumption is realistic when the time it takes for the adversarial agent to attack new inputs and outputs is large compared to the time scale of the system.
We now precisely define the main problem we tackle in this paper.

Problem 2 (Secure state estimation) *For the linear system defined by* (6) *under the attack model defined by* (7)*, what are necessary and sufficient conditions under which the state of the compromised system* (6) *can be reconstructed with bounded delay?*

It is well-known that the secure state estimation problem, when only outputs are under adversarial attacks, is combinatorial and belongs to the class of *NP-hard* problems [31,35]. Therefore we are motivated to design an observer that harness the complexity of this problem.

Problem 3 (Secure observer design) *Assumming conditions in Problem 2 are satsified, how can we design an observer that reconstructs the state of the compromised system?*

3 Conditions for Secure State Estimation

In this section, we solve Problem 2, i.e., we provide conditions on the system described by (6) under which state reconstruction (with bounded delay) is possible. We first develop the notion of sparse strong observability. This section concludes with Theorem 8 that relates this notion to the solution of Problem 2.
In the absence of attacks, the problem of estimating the state of a system while some of the inputs are unknown has been studied and the notion of strong observability was introduced in the literature [16]. For strongly observable systems, it is possible to estimate the state of the system without the knowledge of inputs. The following definition formalizes this concept.

Definition 4 (Strong observability) *An LTI system is called strongly observable if for any initial state $x(0) \in \mathbb{R}^n$ and any input sequence $\{u(t) \in \mathbb{R}^m\}_{t=0}^{\infty}$ there exists an integer $\tau \in \mathbb{N} \cup \{0\}$ such that $x(0)$ can be uniquely recovered from $\{y(t)\}_{t=0}^{\tau}$.*

Note that τ is always upper-bounded by the order of the system. Linearity implies the following lemma.

Lemma 5 *An LTI system is strongly observable if and only if $y(t) = 0\ \forall t \in \mathbb{N} \cup \{0\}$ implies that $x(0) = 0$.*

PROOF. Please refer to Appendix.

It is straightforward to conclude the following corollary.

Corollary 6 *An LTI system is not strongly observable if and only if there exist a non-zero intial state and an input sequence such that $y(t) = 0$ for $t \in \mathbb{N} \cup \{0\}$.*

PROOF. Follows directly from Lemma 5.

It is well-understood that when the adversary is restricted to attacking outputs, state reconstruction is possible only if there is enough redundancy in the outputs of the system. This redundancy can be stated in terms of observability of the system while removing a number of outputs. This property has been formalized in [11] and is called sparse observability [36]. By analogy with sparse observability, we define the notion of (r,s)-sparse strong observability as follows:

Definition 7 ((r,s)-sparse strong observability) *An LTI system (A,B,C,D) with m inputs and p outputs is (r,s)-sparse strongly observable if for any $\Gamma_u \subseteq \{1,\dots,m\}$ and $\Gamma_y \subseteq \{1,\dots,p\}$ with $|\Gamma_u| \leq r$ and $|\Gamma_y| \geq p-s$, the system $(A, B_{(.,\Gamma_u)}, C_{(\Gamma_y,.)}, D_{(\Gamma_y,\Gamma_u)})$ is strongly observable.*

Note that in Definition 7, the value of r and s are upper bounded by the number of inputs and outputs, respectively. This modified notion of strong observability is the key for formalizing redundancy across inputs and outputs. We show that a necessary and sufficient condition for secure state estimation can be stated using this property. Note that $(0,s)$-sparse strong observability is equivalent to the notion of s-sparse observability that was introduced before in the literature [11, 23, 36]. The following theorem is the main theoretical result in this paper.

Theorem 8 *Let the number of attacked inputs and outputs be bounded by r and s, respectively. Under the attack model* (7)*, the state can be reconstructed (possibly with delay) if and only if the underlying system is $(2r,2s)$-sparse strongly observable.*

Remark 9 *It is worth mentioning that the maximum number of attacked outputs, s, cannot be greater than $\lfloor \frac{p}{2} \rfloor$ and it is an inherent limitation of LTI systems with p outputs [11]. However the maximum number of attacked inputs is not inherently restricted by $\lfloor \frac{m}{2} \rfloor$ and can take values up to m, depending on the specific system under the consideration.*

Remark 10 *Pasqualetti et. al. [31] addressed the problem of attack detection and identification in the presence of adversarial inputs and outputs for continuous-time LTI systems. They showed that attack identification is possible if and if for any $\Gamma_u \subseteq \{1,\dots,m\}$ and $\Gamma_y \subseteq \{1,\dots,p\}$ with $|\Gamma_u| \leq 2r$ and $|\Gamma_y| \geq p-2s$, the system $(A, B_{(.,\Gamma_u)}, C_{(\Gamma_y,.)}, D_{(\Gamma_y,\Gamma_u)})$ does not have any invariant zeros.*

It is clear that from the state and the dynamics of the system, the attack can be identified, therefore the attack identification comes free with the solution to the secure estimation problem. Strongly observable LTI systems do not have any invariant zeros (see, for example Theorem 1.8 in [16]). Therefore this theorem shows that under this sparse-attack model, the conditions for identifying the attack also enable one to reconstruct the state, i.e., characterizations of attack identifiability and secure state estimation are equivalent for LTI systems. Putting these together, secure state estimation also comes with the solution to the attack identification problem. However, we provide a direct proof that does not require this machinery.

PROOF. First we show that $(2r,2s)$-sparse strong observability is a sufficient condition for correctly estimating the state. For the sake of the contradiction, assume that the state cannot be reconstructed, i.e., there exist two different (initial) states, denoted by $x^{(1)}$ and $x^{(2)}$, that cannot be distinguished under this attack model. More precisely, there exist two attack strategies that will lead to the same exact (observed) trajectories. We reserve superscripts $.^{(1)}$ and $.^{(2)}$ for variables across those scenarios. Let us denote the adversarial additive terms by $\{w^{(1)}(t)\},\{a^{(1)}(t)\}$ and $\{w^{(2)}(t)\},\{a^{(2)}(t)\}$. We represent the corresponding inputs and outputs of the system by $\{u_S^{(1)}(t)\},\{y_S^{(1)}(t)\}$ and $\{u_S^{(2)}(t)\},\{y_S^{(2)}(t)\}$, and the common (corrupted) measured output and the controller input sequences are denoted by $\{y(t)\}$ and $\{u(t)\}$, respectively.
By the assumption of the attack model (7), there exist $\Gamma_u^{(i)},\overline{\Gamma}_y^{(i)}$ for $i\in\{1,2\}$ with bounded cardinality such that

$$\text{supp}(\{w^{(i)}(t)\})\subseteq\Gamma_u^{(i)},\text{supp}(\{a^{(i)}(t)\})\subseteq\overline{\Gamma}_y^{(i)}, \tag{8}$$

for $i\in\{1,2\}$. Note that

$$\begin{cases} u_S^{(1)}(t)=u(t)+w^{(1)}(t)\\ u_S^{(2)}(t)=u(t)+w^{(2)}(t)\end{cases}, \tag{9}$$

where $u(t)$ is the controller designed input. Therefore

$$\begin{aligned}\text{supp}(\{u_S^{(1)}(t)-u_S^{(2)}(t)\})&=\text{supp}(\{w^{(1)}(t)-w^{(2)}(t)\})\\&\subseteq\Gamma_u^{(1)}\cup\Gamma_u^{(2)}.\end{aligned} \tag{10}$$

Similarly, it is straightforward to conclude that $\text{supp}(\{y_S^{(1)}(t)-y_S^{(2)}(t)\})\subseteq\overline{\Gamma}_y^{(1)}\cup\overline{\Gamma}_y^{(2)}$. We are ready to reach the contradiction. The underlying system is LTI, thus the input sequence $\{u_S^{(1)}(t)-u_S^{(2)}(t)\}$ with the initial state $x^{(1)}-x^{(2)}$ generates the output sequence $\{y_S^{(1)}(t)-y_S^{(2)}(t)\}$. The underlying system is $(2r,2s)$-sparse strongly observable so the sub-system $(A,B_{(.,\Gamma_u)},C_{(\Gamma_y,.)},D_{(\Gamma_y,\Gamma_u)})$ is strongly observable for any $|\Gamma_u|=2r$ and $|\Gamma_y|=p-2s$. Let us choose Γ_u and Γ_y as any set of $2r$ inputs and $p-2s$ outputs such that,

$$\Gamma_u^{(1)}\cup\Gamma_u^{(2)}\subseteq\Gamma_u,\quad \Gamma_y\subseteq\Gamma_y^{(1)}\cap\Gamma_y^{(2)}. \tag{11}$$

Note that $\{y_S^{(1)}(t)|_{\Gamma_y}-y_S^{(2)}(t)|_{\Gamma_y}\}$ is a zero sequence, hence by Lemma 5 we conclude that the corresponding initial state $(x^{(1)}-x^{(2)})$ is zero, which contradicts the assumption of $x^{(1)}\neq x^{(2)}$. Now we prove that $(2r,2s)$-sparse strongly observability is a necessary condition. For the sake of contradiction, suppose that the system described by (6) is not $(2r,2s)$-sparse strongly observable, however, reconstructing the state (possibly with delays) is still possible. We construct two system trajectories with different (initial) states that have exactly the same input and output sequences under suitable attack strategies (additive terms). This implies that estimating the correct state is indeed impossible thereby establishing the desired contradiction.
By the assumption of the contradiction, the underlying system is not $(2r,2s)$-sparse strongly observable, so there exist subsets of inputs and outputs denoted by Γ_u with $|\Gamma_u|=2r$ and Γ_y with $|\Gamma_y|=p-2s$, respectively, such that $(A,B_{(.,\Gamma_u)},C_{(\Gamma_y,.)},D_{(\Gamma_y,\Gamma_u)})$ is not strongly observable. Corollary 6 implies that there exist an initial condition Δx and an input sequence $\{\Delta u(t)\}$ (with its support lying inside Γ_u) that generates an output sequence $\{\Delta y(t)\}$ with $\text{supp}(\{\Delta y(t)\})\subseteq\overline{\Gamma}_y$. One can rewrite $\Delta u(t)$ and $\Delta y(t)$ as sum of two sparse signals, more precisely:

$$\Delta u(t)=\Delta u^{(1)}(t)+\Delta u^{(2)}(t), \tag{12}$$
$$\Delta y(t)=\Delta y^{(1)}(t)+\Delta y^{(2)}(t), \tag{13}$$

where cardinality of $\text{supp}(\{\Delta u^{(i)}(t)\})$ and $\text{supp}(\{\Delta y^{(i)}(t)\})$ are upper-bounded by r and s for $i\in\{1,2\}$, respectively. For example, we can rewrite $\overline{\Gamma}_y=\overline{\Gamma}_y^{(1)}\cup\overline{\Gamma}_y^{(2)}$ where $|\overline{\Gamma}_y^{(i)}|\leq s$ for $i\in\{1,2\}$. Then we define

$$\begin{cases}\Delta y^{(i)}(t)|_{\overline{\Gamma}_y^{(i)}} &:=\Delta y(t)|_{\overline{\Gamma}^{(i)}}\\ \Delta y^{(i)}(t)|_{\Gamma_y^{(i)}} &:=0\end{cases},\quad \text{for}\quad i\in\{1,2\}.$$

Now consider the following two different trajectories of the system

$$\begin{cases}u_S^{(1)}(t) &=\Delta u(t)\\ y_S^{(1)}(t) &=\Delta y(t)\end{cases},\quad \begin{cases}u_S^{(2)}(t) &=0\\ y_S^{(2)}(t) &=0\end{cases} \tag{14}$$

with their initial states

$$\begin{cases}x^{(1)}(0) &=\Delta x\\ x^{(2)}(0) &=0\end{cases}, \tag{15}$$

and their corresponding attack strategies,

$$\begin{cases} w^{(1)}(t) &= \Delta u^{(1)}(t) \\ a^{(1)}(t) &= -\Delta y^{(1)}(t) \end{cases}, \quad \begin{cases} w^{(2)}(t) &= -\Delta u^{(2)}(t) \\ a^{(2)}(t) &= \Delta y^{(2)}(t) \end{cases}. \quad (16)$$

It is straightforward to verify that $\{y^{(1)}(t)\} = \{y^{(2)}(t)\}$ and $\{u^{(1)}(t)\} = \{u^{(2)}(t)\}$, i.e., under the attack model (7) the controlled inputs and the observed outputs are exactly the same for both trajectories while having different states, therefore the proof is complete.

4 Secure Observer Design

In this section, we seek solutions to Problem 3. In the first part, we explain the intuition behind the proposed algorithm that estimates the state despite attacks on inputs and outputs. We give formal guarantees that the algorithm reconstructs the state correctly. In the second part, we introduce the observer by leveraging the SMT paradigm followed by two methods that enhance the run time of state estimation.
Based on the attack model (7), the input to the system is decomposed into two additive terms, the controller-designed input $u(t)$ and the adversarial input $w(t)$. The underlying system (6) is linear and therefore we can easily exclude the effect of the controller-designed input from the output by subtracting its effect. Hence, without loss of generality we assume that the true $u(t)$ is zero.
The proposed algorithm is based on the following proposition.

Proposition 11 *Suppose the underlying system is $(2r,2s)$-sparse strongly observable, and the number of attacked inputs and outputs are bounded by r and s, respectively. Given any subset of inputs and outputs denoted by Γ_u and Γ_y with $|\Gamma_u| \leq r$ and $|\Gamma_y| \geq p-s$, the first statement below implies the second:*

(1) There exist $\hat{\mathbf{U}} \in \mathbb{R}^{n|T|}$ and $\hat{x} \in \mathbb{R}^n$ such that

$$\mathbf{Y}|_{\Gamma_y}(t) = \mathscr{O}_{\Gamma_y}\hat{x} + \mathscr{N}_{\Gamma_u \to \Gamma_y}\hat{\mathbf{U}}. \quad (17)$$

(2) The estimated state $\hat{x}$, is equal to the actual state of the system at time $t-n+1$, $x(t-n+1)$, where n is the order of the underlying system.

Remark 12 *The underlying system is $(2r,2s)$-sparse strongly observable therefore $(A, B_{(.,\Gamma_u)}, C_{(\Gamma_y,.)}, D_{(\Gamma_y,\Gamma_u)})$ is strongly observable. If (17) has a solution, then $\hat{x}$ would be the unique solution for x (see section III-B of [44]).*

PROOF. Let us denote the set of attack-free outputs and under-attack inputs by Γ_y^* and Γ_u^*. At most s outputs are under attack, therefore $|\Gamma_y \cap \Gamma_y^*| \geq p-2s$. Note that $\mathbf{Y}|_{\Gamma_y \cap \Gamma_y^*}$ can be written as follows:

$$\begin{aligned} \mathbf{Y}|_{\Gamma_y \cap \Gamma_y^*} =& O_{\Gamma_y \cap \Gamma_y^*} x(t-n+1) \\ &+ \mathscr{N}_{\Gamma_u \to \Gamma_y \cap \Gamma_y^*} \mathbf{W}|_{\Gamma_u} + \mathscr{N}_{\Gamma_u^* \setminus \Gamma_u \to \Gamma_y \cap \Gamma_y^*} \mathbf{W}|_{\Gamma_u^* \setminus \Gamma_u}. \quad (18) \end{aligned}$$

On the other hand, we can rewrite (17) by taking only outputs in $\Gamma_y \cap \Gamma_y^*$,

$$\mathbf{Y}|_{\Gamma_y \cap \Gamma_y^*} = O_{\Gamma_y \cap \Gamma_y^*}\hat{x} + \mathscr{N}_{\Gamma_u \to \Gamma_y \cap \Gamma_y^*}\hat{\mathbf{U}} + \mathscr{N}_{\Gamma_u^* \setminus \Gamma_u \to \Gamma_y \cap \Gamma_y^*}\mathbf{0}, \quad (19)$$

where $\mathbf{0}$ is a zero vector with appropriate dimensions. The underlying system is $(2r,2s)$-sparse strongly observable, therefore we conclude that the sub-system $\hat{S} := (A, B_{(.,\Gamma_u \cup \Gamma_u^*)}, C_{(\Gamma_y \cap \Gamma_y^*,.)}, D_{(\Gamma_y \cap \Gamma_y^*, \Gamma_u \cup \Gamma_u^*)})$ is strongly observable. One can reinterpret both equations as two (possibly different) valid trajectories of the system $\hat{S}$ that share the same output sequence. Strong observability of $\hat{S}$ implies that $\hat{x} = x(t-n+1)$ which completes the proof.

The main algorithm in this paper builds upon this proposition. We search for a set of inputs and outputs that satisfies equality (17), i.e., we check if there exist $\hat{\mathbf{U}}$ and $\hat{x}$ that make equality (17) hold. Based on Proposition 11, we define a consistency check as follows,

TEST 1 (Consistency Check) *Given subsets of inputs and outputs denoted by Γ_u and Γ_y, TEST(Γ_u, Γ_y) returns true if*

$$\min_{\hat{\mathbf{U}}, \hat{x}} \|\mathbf{Y}|_{\Gamma_y} - \mathscr{O}_{\Gamma_y}\hat{x} - \mathscr{N}_{\Gamma_u \to \Gamma_y}\hat{\mathbf{U}}\| \leq \varepsilon, \quad (20)$$

where $\varepsilon > 0$ is the solver tolerance, due to numerical errors. However, for the sake of clarity, we focus in this paper on the case when ε is negligible [2].

Finding the right subset of inputs and ouputs that satisfies this test is a combinatorial problem in nature and requires exhaustive search. It is well-known that secure state estimation under this attack model is in general *NP-hard* [31, 35]. This test is depicted in Algorithm 2.
In the rest of this section, we introduce an architecture for our observer followed by methods to improve its computational performance. For each input (output), we assign a binary variable $\mathbf{b}_i \in \mathbb{B}$ ($\mathbf{c}_i \in \mathbb{B}$) that indicates if the corresponding input (output) is under attack or not, i.e., $\mathbf{b}_i = 1$ ($\mathbf{c}_i = 1$) if the i^{th} input (output) is under attack. In the rest of this paper, we use the bold letters ($\mathbf{b}$ and $\mathbf{c}$) to denote these Boolean variables and we reserve non-bold type face (b and c) as instances of them. Finding the right assignment of these Boolean variables is combinatorial in nature and in order to efficiently decide which set of inputs and outputs satisfies the TEST in (20), we design an observer using the lazy SMT paradigm [4].

[2] Note that the minimum always exists for (20) as the cost function is a semi-definite quadratic function.

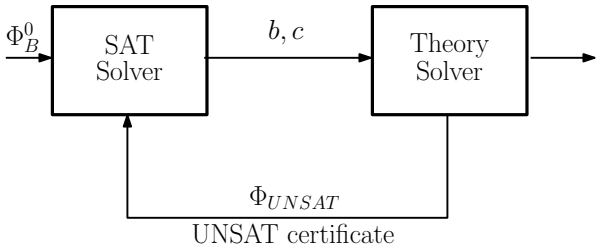

Fig. 2. The lazy SMT paradigm.

4.1 Overall Architecture

The observer consists of two blocks that interact with each other, a propositional satisfiability (SAT) solver and a theory solver. The former reasons about the combination of Boolean and pseudo-Boolean constraints and produces a feasible instance of $\mathbf{b} \in \mathbb{B}^m$ and $\mathbf{c} \in \mathbb{B}^p$, based on its current state. The theory solver checks the consistency of Boolean variables using the consistency test, and when the test fails, it encodes the inconsistency as a pseudo-Boolean constraint and returns it to the SAT solver. The general architecture is depicted in Figure 2.

The initial pseudo-Boolean constraint only bounds the number of attacked inputs and outputs, i.e.,

$$\Phi_B := (\sum_{i=1}^{m} \mathbf{b}_i \leq r) \bigwedge (\sum_{j=1}^{p} \mathbf{c}_i \leq s). \tag{21}$$

Initially, the SAT solver generates instances of $\mathbf{b}$ and $\mathbf{c}$ that satisfy Φ_B. The theory solver checks whether $\Gamma_u := \text{supp}(b)$ and $\Gamma_y := \overline{\text{supp}(c)}$ satisfies the consistency check. If the test is satisfied, then the algorithm terminates and returns the (delayed) estimate of the state. Otherwise, the theory solver outputs UNSAT and generates a reason for the conflict, a certificate, or a counterexample that is denoted by Φ_{cert}. This counterexample encodes the inconsistency among the chosen inputs and outputs. The following always constitutes a naive certificate.

$$\Phi_{\text{naive-cert}} := \sum_{i \in \overline{\text{supp}(b)}} \mathbf{b}_i + \sum_{j \in \overline{\text{supp}(c)}} \mathbf{c}_j \geq 1. \tag{22}$$

On the next iteration, the SAT solver updates the constraint by conjoining Φ_{cert} to Φ_B, and generates another feasible assignment for $\mathbf{b}$ and $\mathbf{c}$. This procedure is repeated until the theory solver returns SAT as illustrated in Algorithm 1.

Note that Proposition 11 implies that the SAT solver eventually produces an assignment that satisfies the consistency test and therefore Algorithm 1 always terminates. The size of the certificate plays an important role in the overall execution time of the algorithm [35]. Note that the attack model considered in [35] is restricted to outputs, and the major contribution of our work is to handle both input and output attacks. In the next section, we focus on constructing shorter counterexamples to improve the run time.

Algorithm 1. Secure state estimator
Require: A, B, C, D (system), Y (output), r, s (bounds)
1: status $\leftarrow$ UNSAT
2: $\Phi_{\text{cert}} \leftarrow$ True
3: $\Phi_B \leftarrow (\sum_{i \in \{1,\ldots,m\}} \mathbf{b}_i \leq r) \bigwedge (\sum_{i \in \{1,\ldots,p\}} \mathbf{c}_i \leq s)$
4: **while** status = UNSAT **do**
5: $\quad \Phi_B \leftarrow \Phi_B \bigwedge \Phi_{\text{cert}}$
6: $\quad (b, c) \leftarrow$ SAT-solver(Φ_B)
7: $\quad$ (status, x) $\leftarrow$ T-solver.check(supp$(b), \overline{\text{supp}(c)}$)
8: $\quad \Phi_{\text{cert}} \leftarrow$ T-solver.Certificate(supp$(b), \overline{\text{supp}(c)}$)
9: **return** (x, b, c)

Algorithm 2. T-solver.check
Require: Γ_u, Γ_y
1: **Solve:** $(\hat{x}, \hat{\mathbf{U}}) = \text{argmin}_{x,\mathbf{U}} \|\mathbf{Y}|_{\Gamma_y} - \mathscr{O}_{\Gamma_y} x - \mathscr{N}_{\Gamma_u \to \Gamma_y} \mathbf{U}\|$
2: **if** $\|\mathbf{Y}|_{\Gamma_y} - \mathscr{O}_{\Gamma_y} \hat{x} - \mathscr{N}_{\Gamma_u \to \Gamma_y} \hat{\mathbf{U}}\| \leq \varepsilon$ **then**
3: $\quad$ status = SAT
4: **else**
5: $\quad$ status = UNSAT
6: **return** $(\text{status}, \hat{x})$

4.2 SAT certificate

In this part, we improve the efficiency of Algorithm 1 by constructing a shorter certificate (counter-example or conflicts). As it was discussed before, the naive certificate only excludes the current assignment of $\mathbf{b}$ and $\mathbf{c}$ from the search space of the SAT solver, however, by exploiting the structure of the underlying system, we show that we can further decrease the size of the certificate and therefore prune the search space more efficiently.

One of the main results of this paper is to show that we can always find a smaller conflicting subset of inputs and outputs. We propose two methods for generating shorter certificates. The first method reduces the size of the counterexample by at least $s-1$, we explain this method in Lemma 13 and give a formal proof of the existence of such shorter certificate. In practice, however we observe the reduction in the length of conflicts is much larger than this theoretical bound. The second method is inspired by the QUICKXPLAIN algorithm. This method generates counter-examples that are irreducible, meaning that we cannot reduce the size of the counter-example by removing some of it's entries. We also note that by generating multiple certificates at each iteration we can further enhance the execution time. At the end of this section Lemma 15 states that for a generic LTI system the size of the certificate cannot be smaller than $m+1$.

Let us assume that the SAT solver hypothesized $\Gamma_u^{\text{SAT}} := \text{supp}(b)$ and $\Gamma_y^{\text{SAT}} := \overline{\text{supp}(c)}$ as the set of compromised inputs and safe outputs, respectively. The main intuition behind both methods is to look for $\Gamma_u^{\text{cert}} \supseteq \Gamma_u^{\text{SAT}}$ and $\Gamma_y^{\text{cert}} \subseteq \Gamma_y^{\text{SAT}}$ that would not satisfy the consistency test. Note that the certificate consists of inputs in $\overline{\Gamma}_u^{\text{cert}}$ and outputs in Γ_y^{cert}.

Algorithm 3. T-solver.Certificate 1
Require: $\Gamma_u^{\text{SAT}}, \Gamma_y^{\text{SAT}}$
step 1: Conduct a linear search in the input set
Sort $\overline{\Gamma}_u^{\text{SAT}}$
status $\leftarrow$ UNSAT, $j \leftarrow \emptyset, \Gamma_u^{\text{cert}} \leftarrow \Gamma_u^{\text{SAT}}$
while status == UNSAT **and** $|\Gamma_u^{\text{cert}}| < 2r$ **do**
  $\Gamma_u^{\text{cert}} \leftarrow \Gamma_u^{\text{cert}} \cup \{j\}$
  pick another input $j \in \overline{\Gamma}_u^{\text{SAT}}$
  $(\text{status}, x) \leftarrow$ T-Solver.check$(\Gamma_u^{\text{cert}} \cup \{j\}, \Gamma_y^{\text{SAT}})$
step 2: Conduct a linear search in the output set
Sort Γ_y^{SAT}
Pick a subset of size $p-2s$: $\Gamma_y^{\text{temp}} \subseteq \Gamma_y^{\text{SAT}}$
status $\leftarrow$ SAT, $i \leftarrow \emptyset$
while status == SAT **do**
  $\Gamma_y^{\text{cert}} \leftarrow \Gamma_y^{\text{temp}} \cup \{i\}$
  $(\text{status}, x) \leftarrow$ T-Solver.check$(\Gamma_u^{\text{cert}}, \Gamma_y^{\text{cert}})$
  Pick another output $i \in \Gamma_y^{\text{SAT}} \setminus \Gamma_y^{\text{temp}}$
$\Phi_{\text{cert}}^1 \leftarrow \sum_{j \in \overline{\Gamma}_u^{\text{cert}}} \mathbf{b}_j + \sum_{i \in \Gamma_y^{\text{cert}}} \mathbf{c}_i \geq 1$
return Φ_{cert}^1

4.3 Method I based on heuristics

Method I reduces the size of the certificate by increasing the size of (supposedly under attack) inputs (Γ_u^{cert}) followed by decreasing the size of (supposedly safe) outputs (Γ_u^{cert}). The summary of the above procedure of shortening certificates is illustrated in Algorithm 3. We begin by adding inputs to Γ_u^{SAT} while making sure TEST still returns false and the number of inputs is bounded by $2r$. Let us denote this new set of inputs by Γ_u^{cert}.
At the second step, we shrink the set of conflicting outputs in order to further shorten the size of the counterexample. Let us denote a subset of Γ_y^{SAT} of size $p-2s$ by Γ_y^{temp}. The following lemma shows we can reduce the size of conflicting outputs at least by $s-1$.

Lemma 13 *Assume that the system S is $(2r,2s)$-sparse strongly observable, and the number of attacked inputs and outputs are bounded by r and s, respectively. Pick any subset of inputs and outputs denoted by Γ_u^{cert} and Γ_y^{SAT} with $|\Gamma_u^{cert}| \leq 2r$ and $|\Gamma_y^{SAT}| \geq p-s$, that do not satisfy the consistency check* (20)*. Given any subset of at most $p-2s$ outputs denoted by $\Gamma_y^{temp} \subseteq \Gamma_y^{SAT}$, one of the following is true:*

(1) TEST($\Gamma_u^{cert}, \Gamma_y^{temp}$) returns false,
(2) There exists an output $i \in \Gamma_y^{SAT} \setminus \Gamma_y^{temp}$ such that TEST($\Gamma_u^{cert}, \Gamma_y^{temp} \cup \{i\}$) returns false.

PROOF. Please refer to Appendix.

We denote this smaller set of conflicting outputs Γ_y^{temp}(if TEST($\Gamma_u^{\text{cert}}, \Gamma_y^{\text{temp}}$) returns false, otherwise $\Gamma_y^{\text{temp}} \cup \{i\}$) by Γ_y^{cert}. Lemma 13 gives formal guarantees of the existence of shorter certificates which hold no matter how the subsets of inputs and outputs (Γ_u^{temp} and Γ_y^{temp}) are chosen. This lemma shows that Method I reduces the size of the certificate by at least $s-1$.
In practice, we choose these subsets based on heuristics that have for objective a decrease in the overall running time. We assign slack variables to inputs and outputs similarly to [35] and [37], and sort them based on the structure of the system. Recall that Algorithm 3 shortens the certificate by reducing the number of inputs followed by the reduction in the number of outputs, i.e., we *simultaneously* reducing both inputs and outputs in the certificate. We observe that by generating two counterexamples, we can prune the search space of the SAT solver more efficiently. Similarly to Algorithm 5, we can find two counterexamples by reducing the number of inputs following a reduction in the number of outputs and vice-verse.
Sorting $\overline{\Gamma}_u^{\text{SAT}}$ and Γ_y^{SAT}:
Assuming TEST($\Gamma_u^{\text{SAT}}, \Gamma_y^{\text{SAT}}$) returns false, we assign slack variables to inputs in $\overline{\Gamma}_u^{\text{SAT}}$ and outputs in Γ_y^{SAT}, denoted by $\text{slack}_u(j)$ and $\text{slack}_y(i)$, respectively. Let us denote a solution to the optimization (20) inside TEST($\Gamma_u^{\text{SAT}}, \Gamma_y^{\text{SAT}}$) by $\hat{x}$ and $\hat{\mathbf{U}}$.
We define $\text{slack}_u(j)$ for $j \in \overline{\Gamma}_u^{\text{SAT}}$ as the norm of the projection of $\mathbf{Y}|_{\Gamma_y^{\text{SAT}}} - \mathscr{O}_{\Gamma_y^{\text{SAT}}}\hat{x} - \mathscr{N}_{\Gamma_u^{\text{SAT}} \to \Gamma_y^{\text{SAT}}}\hat{\mathbf{U}}$ onto the column space of $\mathscr{N}_{j \to \Gamma_y^{\text{SAT}}}$,

$$\text{slack}_u(j) := \|\mathscr{N}_{j \to \Gamma_y^{\text{SAT}}} \mathscr{N}_{j \to \Gamma_y^{\text{SAT}}}^{\dagger} \left(\mathbf{Y}|_{\Gamma_y^{\text{SAT}}} - \mathscr{O}_{\Gamma_y^{\text{SAT}}}\hat{x} - \mathscr{N}_{\Gamma_u^{\text{SAT}} \to \Gamma_y^{\text{SAT}}}\hat{\mathbf{U}} \right) \|. \tag{23}$$

This slack variable measures how much of the residual can be justified by considering j in addition to Γ_u^{SAT}. Note that we want to append inputs to Γ_u^{SAT} while having a false TEST. We first normalize these slack variables by the norm of the corresponding invertibility matrix, and $\overline{\Gamma}_u^{\text{SAT}}$ is obtained by sorting slack variables in *ascending* order.
We define $\text{slack}_y(i)$ as the residual of each output:

$$\text{slack}_y(i) := \|\mathbf{Y}|_i - \mathscr{O}_i\hat{x} - \mathscr{N}_{\Gamma_u^{\text{SAT}} \to \{i\}}\mathbf{U}\|, \quad i \in \Gamma_y^{\text{SAT}}. \tag{24}$$

Note that,

$$\sum_{i \in \Gamma_u^{\text{SAT}}} \text{slack}_y(i) = \min_{\hat{\mathbf{U}}, \hat{x}} \|\mathbf{Y}|_{\Gamma_y^{\text{SAT}}} - \mathscr{O}_{\Gamma_y^{\text{SAT}}}\hat{x} - \mathscr{N}_{\Gamma_u^{\text{SAT}} \to \Gamma_y^{\text{SAT}}}\hat{\mathbf{U}}\|. \tag{25}$$

We first normalize each slack variable by the norm of the corresponding observability matrix. Recall that we aim to find a smaller subset of Γ_u^{SAT} while ensuring TEST returns false. We pick the output with the highest slack variable as the first element of Γ_u^{SAT}. We sort the rest based on the dimension of the kernel of each observability matrix, following the intuition provided in [35].

4.4 Method II based on QuickXplain

The second method (Algorithm 5) is inspired by QUICKXPLAIN and generates a counter-example by pruning the naive-certificate (22) to make it irreducible. We formally define this property as follows,

Definition 14 (Irreducible certificate) *A certificate consisting of inputs $\overline{\Gamma}_u$ and outputs Γ_y is irreducible, if no other subset of it can generate a conflict, i.e., for all subsets denoted by $\overline{\Gamma}'_u \subseteq \overline{\Gamma}_u$ and $\Gamma'_y \subseteq \Gamma_y$ the following are equivalent:*

(1) $\overline{\Gamma}'_u$ and Γ'_y generate a conflict.
(2) $\overline{\Gamma}'_u = \overline{\Gamma}_u$ and $\Gamma'_y = \Gamma_y$.

One cannot prune irreducible certificates and each element is necessary for the set to remain a counter-example. Let Δ^{SAT} be the elements (consisting of inputs $\overline{\Gamma}_u^{\text{SAT}}$ and outputs Γ_y^{SAT}) of the naive certificate. For ease of exposition we slightly abuse notation to denote $\text{TEST}(\Gamma_u^{\text{SAT}}, \Gamma_y^{\text{SAT}})$ by $\text{TEST}(\Delta^{\text{SAT}})$. We denote the output of this algorithm by Δ_{cert} which consists of inputs $\overline{\Gamma}_u^{\text{cert}}$ and outputs Γ_y^{cert}.
This method consists of an exploration phase in which it finds an element (input or output) that belongs to an irreducible certificate. Let us denote an enumeration of Δ^{SAT} by $e_1, \cdots, e_k$, and the internal state by $\Delta_{\text{temp}} \leftarrow \emptyset$. This method begins by adding step-by-step elements of Δ^{SAT} to Δ_{temp}. The first element ($e_i \in \Delta^{\text{SAT}}$) that fails $\text{TEST}(\Delta_{\text{temp}})$ is part of an irreducible certificate, and therefore is added to Δ_{cert}. In order to find further elements of this certificate, we keep e_i in the background and the first element that fails the consistency check is added to Δ_{cert}. This repeated process can be implemented efficiently by using the divide and conquer paradigm as depicted in Algorithm 4. When an element e_i of Δ^{SAT} is detected we divide the the remaining elements into two disjoint subsets $\Delta^1 := \{e_1, \cdots, e_j\}$ and $\Delta^2 := \{e_{j+1}, \cdots, e_{i-1}\}$. We can now recursively apply the algorithm to find a conflict Δ^2_{cert} among Δ^2 by keeping the set Δ^1 in the background and a conflict Δ^1_{cert} among Δ^1 by keeping the set Δ^2_{cert} in the background. This method of finding an irreducible subset is depicted in Algorithm 4
Note that the resulting counter-example depends on the initial enumeration of elements in Δ^{SAT}. If the all the inputs (outputs) are ahead of outputs (inputs), then the resulting counter-example mostly consists of inputs (outputs). In order to have the maximal reduction in the search space of the SAT solver at each iteration, we produce three certificate using this method, putting inputs first, outputs first and mixing both inputs and outputs.
In the last part of this section, we look at the certificate size for a generic LTI system. We observe that the certificate size cannot be smaller that the number of inputs which is stated formally in the following lemma.

Lemma 15 *For a generic LTI system the size of the certificate is always lower bounded by $m+1$, where m is the number of inputs.*

Algorithm 4. T-solver.QuickXplain
Require: $\Delta^0_{\text{cert}}, \Delta^0$
1: **if** T-solver.check(Δ^0_{cert}) = UNSAT or $\Delta^0 == \emptyset$ **then**
2: **return** $\emptyset$
Let $e_1, \cdots, e_k$ be an enumeration of Δ^0
3: $i \leftarrow 0$, $\Delta_{\text{temp}} \leftarrow \Delta^0_{\text{cert}}$,
4: **while** T-solver.check(Δ_{temp}) = SAT and $i \leq k$ **do**
5: $i \leftarrow i+1$
6: $\Delta_{\text{temp}} \leftarrow \Delta_{\text{temp}} \cup e_i$
7: $\Delta^i_{\text{cert}} \leftarrow \Delta_{\text{temp}}$
8: $\Delta_{\text{cert}} \leftarrow e_i$, $j \leftarrow \lfloor \frac{i}{2} \rfloor$
9: $\Delta^1 \leftarrow \{e_1, \cdots, e_j\}$
10: $\Delta^2 \leftarrow \{e_{j+1}, \cdots, e_{i-1}\}$
11: $\Delta_{\text{cert}} \leftarrow \Delta_{\text{cert}} \cup$ T-solver.QuickXplain($\Delta^j_{\text{cert}} \cup \Delta_{\text{cert}}, \Delta^2$)
12: $\Delta_{\text{cert}} \leftarrow \Delta_{\text{cert}} \cup$ T-solver.QuickXplain($\Delta^0_{\text{cert}} \cup \Delta_{\text{cert}}, \Delta^1$)
13: **return** Δ_{cert}

Algorithm 5. T-solver.Certificate 2
Require: $\Gamma_u^{\text{SAT}}, \Gamma_y^{\text{SAT}}$
1: $\Delta_{\text{cert}} \leftarrow$ T-solver.QuickXplain($\emptyset, \overline{\Gamma}_u^{\text{SAT}} \cup \Gamma_y^{\text{SAT}}$)
2: Divide Δ_{cert} to inputs $\overline{\Gamma}_u^{\text{cert}}$ and outputs Γ_y^{cert}
3: $\Phi^2_{\text{cert}} \leftarrow \sum_{j \in \overline{\Gamma}_u^{\text{cert}}} \mathbf{b}_j + \sum_{i \in \Gamma_y^{\text{cert}}} \mathbf{c}_i \geq 1$
4: **return** Φ^2_{cert}

PROOF. Please refer to Appendix.

5 Simulation Results

We implemented our SMT-based estimator in MATLAB while interfacing with the SAT solver SAT4J [21] and assessed its performance in two case studies, randomly generated LTI systems and a chemical plant. We report the overall running time by using the two proposed methods, Algorithm 3 and Algorithm 5.

5.1 Random Systems

We randomly generate systems with a fixed state dimension ($n = 40$) and increase the number of inputs and outputs. Each system is generated by drawing entries of (A, B, C, D) according to uniform distribution, when necessary we scale A to ensure that the spectral radius is close to one. In each experiment, twenty percent of inputs and outputs are under adversarial attacks, and we generate the support set for the adversarial signals uniformly at random. Attack signals and the initial states are drawn according to independent and normally distributed random variables with zero mean and unit variance. All the systems under experiment satisfy a suitable sparse strong observability condition as described

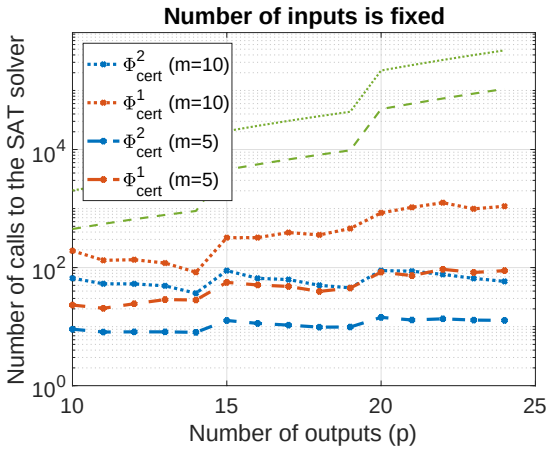

Fig. 3. Number of calls to the SAT solver in Algorithm 1 using Φ^1_{cert}, Φ^2_{cert} versus the number of outputs (p) for a fixed number of inputs. Green dotted and green dashed lines are upper-bounds for the number of the SAT solver calls when using the naive certificate for $m = 5$ and $m = 10$, respectively.

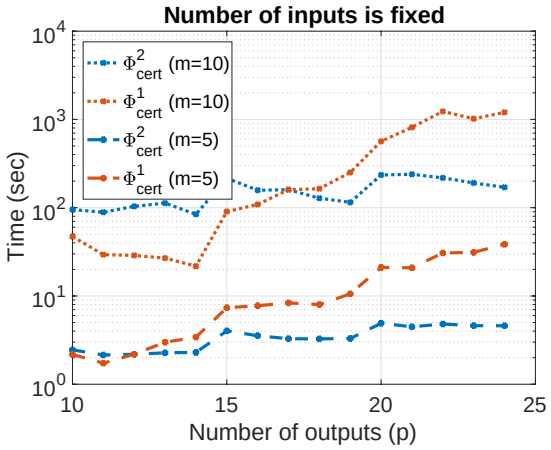

Fig. 4. Execution time of Algorithm 1 using Φ^1_{cert}, Φ^2_{cert} versus the number of outputs (p) and inputs (m).

in Section 3.

Figures 3 and 4 report the results of the simulations, each point represents the average of 20 experiments. All the experiments run on an Intel Core i5 2.7GHz processor with 16GB of RAM. We verify the run-time improvement resulting from using the shorter certificates, Φ^1_{cert} and Φ^2_{cert}, compared to the theoretical upper-bound of the brute-force approach in Figure 3. For instance, consider the scenario with $p = 24$ and $m = 10$ in Figures 3 and 4. In the brute-force approach, we require to check all $\binom{24}{4} \times \binom{10}{2} \approx 4.8 * 10^5$ different combinations of inputs and outputs, however, by exploiting either Φ^1_{cert} or Φ^2_{cert} we observe a substantial improvement. We observe that although Φ^2_{cert} gives a worse run time for systems with smaller number of outputs, it scales better compared to Φ^1_{cert} when the number of inputs and outputs grow.

5.2 *Chemical Plant*

In this part, we use the proposed observer to detect attacks on inputs and outputs of a simplified version of the Tennessee Eastman control challenge problem [9]. Ricker [32] derived a continuous time LTI model of the plant interaction in its steady state. This system consists of 4 control inputs and 10 measured outputs and the linearized model has 8 state variables. The structure of the continuous-time dynamics is reported below.

$$\frac{dx}{dt} = \begin{bmatrix} * & * & * & * & * & * & * & 0 \\ * & * & * & * & * & 0 & * & 0 \\ * & * & * & * & * & 0 & * & 0 \\ * & * & * & * & 0 & 0 & 0 & * \\ 0 & 0 & 0 & 0 & * & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & * & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & * & 0 \\ 0 & 0 & 0 & * & 0 & 0 & 0 & * \end{bmatrix} x + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ * & 0 & 0 & 0 \\ 0 & * & 0 & 0 \\ 0 & 0 & * & 0 \\ 0 & 0 & 0 & * \end{bmatrix} u,$$

$$y = \begin{bmatrix} 0 & 0 & 0 & 0 & * & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & * & 0 & 0 \\ * & * & * & * & 0 & 0 & * & 0 \\ * & * & * & * & 0 & 0 & 0 & * \\ * & * & * & * & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & * & 0 & 0 & 0 & 0 \\ * & * & * & 0 & 0 & 0 & 0 & 0 \\ * & * & * & 0 & 0 & 0 & 0 & 0 \\ * & * & * & 0 & 0 & 0 & 0 & 0 \\ * & * & * & 0 & 0 & 0 & * & * \end{bmatrix} x,$$

where $*$ represents a non-zero entry [3], and $x \in \mathbb{R}^8$, $u \in \mathbb{R}^4$ and $y \in \mathbb{R}^{10}$ are state, input and output variables, respectively. The only known limitation of this LTI model is the system should operate close to its steady-state. We obtain a discrete-time model by discretizing the continuous-time model assuming a zero-order hold for the input u, with a time-step of $5s$. The attacker can read all the inputs and outputs and manipulate one control input and two measured outputs. The linearized system is $(2,4)$-sparse strongly observable, therefore our observer can correctly reconstruct the state under this attack model.

We randomly generate attack signals and the initial state according to independent and normally distributed random variables. The support set of attacks are drawn uniformly at random, and in each experiment one input and two outputs are under adversarial attacks.

The proposed observer in this paper can correctly reconstruct the (delayed) state after 8 samples, and the average performance of 20 experiments, by using Φ^1_{cert} and Φ^2_{cert} is reported in Table 1. The overall execution time is the run time of the observer after receiving all the required samples from the plant, and it does not take the sampling time of the plant into account. We observe that the execution time of the observer to reconstruct the state and to detect attacks is much smaller compared to the sampling time of the plant.

[3] For the exact dynamics of the LTI model, see [32]

Table 1
Average performance of the proposed observer

	Overall execution time	Number of calls to the SAT solver
Φ^1_{cert}	$0.22s$	20.05
Φ^2_{cert}	$0.21s$	7.95

6 Conclusion

In this paper, we considered the problem of secure state estimation when inputs and/or outputs are under adversarial attacks. In this set-up, there is no restriction on how the adversary manipulates inputs and outputs. By introducing the notion of sparse strong observability, we derived necessary and sufficient conditions under which state estimation is possible given bounds on the number of attacked outputs and inputs. Furthermore, we demonstrated the scalability and effectiveness of the proposed estimator with numerical simulations.

References

[1] Saurabh Amin, Galina A Schwartz, and Amir Hussain. In quest of benchmarking security risks to cyber-physical systems. *IEEE Network*, 27(1):19–24, 2013.

[2] Cheng-Zong Bai, Vijay Gupta, and Fabio Pasqualetti. On kalman filtering with compromised sensors: Attack stealthiness and performance bounds. *IEEE Trans. Autom. Control*, 62(12):6641–6648, 2017.

[3] Cheng-Zong Bai, Fabio Pasqualetti, and Vijay Gupta. Data-injection attacks in stochastic control systems: Detectability and performance tradeoffs. *Automatica*, 82:251–260, 2017.

[4] Clark W Barrett, Roberto Sebastiani, Sanjit A Seshia, and Cesare Tinelli. Satisfiability modulo theories. *Handbook of satisfiability*, 185:825–885, 2009.

[5] Mogens Blanke, Michel Kinnaert, Jan Lunze, Marcel Staroswiecki, and J Schröder. *Diagnosis and fault-tolerant control*, volume 691. Springer, 2006.

[6] Alvaro A Cárdenas, Saurabh Amin, and Shankar Sastry. Research challenges for the security of control systems. In *HotSec*, 2008.

[7] Michelle S Chong, Masashi Wakaiki, and Joao P Hespanha. Observability of linear systems under adversarial attacks. In *American Control Conference (ACC)*, pages 2439–2444, 2015.

[8] Claudio De Persis and Pietro Tesi. Input-to-state stabilizing control under denial-of-service. *IEEE Transactions on Automatic Control*, 60(11):2930–2944, 2015.

[9] James J Downs and Ernest F Vogel. A plant-wide industrial process control problem. *Computers & chemical engineering*, 17(3):245–255, 1993.

[10] S. Farahmand, G. B. Giannakis, and D. Angelosante. Doubly robust smoothing of dynamical processes via outlier sparsity constraints. *IEEE Transactions on Signal Processing*, 59(10):4529–4543, Oct 2011.

[11] Hamza Fawzi, Paulo Tabuada, and Suhas Diggavi. Secure estimation and control for cyber-physical systems under adversarial attacks. *IEEE Transactions on Automatic Control*, 59(6):1454–1467, 2014.

[12] Jairo Giraldo, David Urbina, Alvaro Cardenas, Junia Valente, Mustafa Faisal, Justin Ruths, Nils Ole Tippenhauer, Henrik Sandberg, and Richard Candell. A survey of physics-based attack detection in cyber-physical systems. *ACM Computing Surveys (CSUR)*, 51(4):76, 2018.

[13] Andy Greenberg. Hackers remotely kill a jeep on the highway, with me in it. *[online] http://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway*, 2015.

[14] Abhishek Gupta, Cédric Langbort, and Tamer Basar. Optimal control in the presence of an intelligent jammer with limited actions. In *49th IEEE Conference on Decision and Control (CDC)*, pages 1096–1101, 2010.

[15] Farshad Harirchi and Necmiye Ozay. Guaranteed model-based fault detection in cyber-physical systems: A model invalidation approach. *arXiv preprint arXiv:1609.05921*, 2016.

[16] M.L.J. Hautus. Strong detectability and observers. *Linear Algebra and its Applications*, 50(Supplement C):353 – 368, 1983.

[17] Harold Lee Jones. *Failure detection in linear systems.* PhD thesis, Massachusetts Institute of Technology, 1973.

[18] Ulrich Junker. Quickxplain: Conflict detection for arbitrary constraint propagation algorithms. In *IJCAI01 Workshop on Modelling and Solving problems with constraints*, 2001.

[19] Leo Kelion. Nissan leaf electric cars hack vulnerability disclosed. *[online] http://www.bbc.com/news/technology-35642749*, 2016.

[20] Ralph Langner. Stuxnet: Dissecting a cyberwarfare weapon. *IEEE Security & Privacy*, 9(3):49–51, 2011.

[21] Daniel Le Berre and Anne Parrain. The sat4j library, release 2.2, system description. *Journal on Satisfiability, Boolean Modeling and Computation*, 7:59–64, 2010.

[22] J. Mattingley and S. Boyd. Real-time convex optimization in signal processing. *IEEE Signal Processing Magazine*, 27(3):50–61, May 2010.

[23] Shaunak Mishra, Yasser Shoukry, Nikhil Karamchandani, Suhas Diggavi, and Paulo Tabuada. Secure state estimation: Optimal guarantees against sensor attacks in the presence of noise. *IEEE Transactions on Control of Network Systems*, 4(1):49–59, 2017.

[24] Yilin Mo, Rohan Chabukswar, and Bruno Sinopoli. Detecting integrity attacks on scada systems. *IEEE Transactions on Control Systems Technology*, 22(4):1396–1407, 2014.

[25] Yilin Mo, Emanuele Garone, Alessandro Casavola, and Bruno Sinopoli. False data injection attacks against state estimation in wireless sensor networks. In *49th IEEE Conference on Decision and Control (CDC)*, pages 5967–5972, 2010.

[26] Yilin Mo, Tiffany Hyun-Jin Kim, Kenneth Brancik, Dona Dickinson, Heejo Lee, Adrian Perrig, and Bruno Sinopoli. Cyber–physical security of a smart grid infrastructure. *Proceedings of the IEEE*, 100(1):195–209, 2012.

[27] Yilin Mo and Bruno Sinopoli. Secure control against replay attacks. In *47th Annual Allerton Conference on Communication, Control, and Computing*, pages 911–918. IEEE, 2009.

[28] Yilin Mo and Bruno Sinopoli. On the performance degradation of cyber-physical systems under stealthy integrity attacks. *IEEE Transactions on Automatic Control*, 61(9):2618–2624, 2016.

[29] Yorie Nakahira and Yilin Mo. Dynamic state estimation in the presence of compromised sensory data. In *54th Annual Conference on Decision and Control (CDC)*, pages 5808–5813. IEEE, 2015.

[30] Miroslav Pajic, James Weimer, Nicola Bezzo, Paulo Tabuada, Oleg Sokolsky, Insup Lee, and George J Pappas. Robustness of attack-resilient state estimators. In *ICCPS'14: ACM/IEEE 5th International Conference on Cyber-Physical Systems (with CPS Week 2014)*, pages 163–174, 2014.

[31] Fabio Pasqualetti, Florian Dorfler, and Francesco Bullo. Attack detection and identification in cyber-physical systems. *IEEE Transactions on Automatic Control*, 58(11):2715–2729, 2013.

[32] N Lawrence Ricker. Model predictive control of a continuous, nonlinear, two-phase reactor. *Journal of Process Control*, 3(2):109–123, 1993.

[33] Henrik Sandberg and André MH Teixeira. From control system security indices to attack identifiability. In *Science of Security for Cyber-Physical Systems Workshop (SOSCYPS)*, pages 1–6. IEEE, 2016.

[34] Danial Senejohnny, Pietro Tesi, and Claudio De Persis. A jamming-resilient algorithm for self-triggered network coordination. *arXiv preprint arXiv:1603.02563*, 2016.

[35] Yasser Shoukry, Pierluigi Nuzzo, Alberto Puggelli, Alberto L Sangiovanni-Vincentelli, Sanjit A Seshia, and Paulo Tabuada. Secure state estimation for cyber physical systems under sensor attacks: a satisfiability modulo theory approach. *IEEE Transactions on Automatic Control*, 62(10):4917–4932, 2017.

[36] Yasser Shoukry and Paulo Tabuada. Event-triggered state observers for sparse sensor noise/attacks. *IEEE Transactions on Automatic Control*, 61(8):2079–2091, 2016.

[37] M. Showkatbakhsh, Y. Shoukry, H. Chen R, S. Diggavi, and P. Tabuada. An SMT-based approach to secure state estimation under sensor and actuator attacks. In *Decision and Control (CDC), IEEE 56th Conference on*, pages 7177–7182. IEEE, 2017.

[38] Mehrdad Showkatbakhsh, Paulo Tabuada, and Suhas Diggavi. Secure system identification. In *54th Annual Allerton Conference on Communication, Control, and Computing*, pages 1137–1141. IEEE, 2016.

[39] Mehrdad Showkatbakhsh, Paulo Tabuada, and Suhas Diggavi. System identification in the presence of adversarial outputs. In *Decision and Control (CDC), IEEE 55th Conference on*, pages 7177–7182. IEEE, 2016.

[40] Roy S Smith. Covert misappropriation of networked control systems: Presenting a feedback structure. *Control Systems Magazine, IEEE*, 35(1):82–92, 2015.

[41] Shreyas Sundaram, Miroslav Pajic, Christoforos N Hadjicostis, Rahul Mangharam, and George J Pappas. The wireless control network: monitoring for malicious behavior. In *49th IEEE Conference on Decision and Control (CDC)*, pages 5979–5984, 2010.

[42] Ashish Tiwari, Bruno Dutertre, Dejan Jovanović, Thomas de Candia, Patrick D Lincoln, John Rushby, Dorsa Sadigh, and Sanjit Seshia. Safety envelope for security. In *ACM Proceedings of the 3rd international conference on High confidence networked systems*, pages 85–94, 2014.

[43] Sze Zheng Yong, Ming Qing Foo, and Emilio Frazzoli. Robust and resilient estimation for cyber-physical systems under adversarial attacks. In *American Control Conference (ACC), 2016*, pages 308–315. IEEE, 2016.

[44] T Yoshikawa and S Bhattacharyya. Partial uniqueness: Observability and input identifiability. *IEEE Transactions on Automatic Control*, 20(5):713–714, 1975.

[45] Minghui Zhu and Sonia Martinez. On the performance analysis of resilient networked control systems under replay attacks. *IEEE Transactions on Automatic Control*, 59(3):804–808, 2014.

PROOF. [Lemma 5] We first prove the sufficiency part. For the sake of contradiction, suppose that the underlying system is not strongly observable but the property of Corollary 5 is true. If the underlying system (6) is not strongly observable, it means there exist two initial conditions, denoted by $x^{(1)}(0)$ and $x^{(2)}(0)$ possibly with different input sequences denoted by $\{u^{(1)}(t)\}$ and $\{u^{(2)}(t)\}$, respectively, that correspond to the same output sequence $\{y(t)\}$. The underlying system is linear, therefore the nonzero initial condition of $x^{(1)}(0)-x^{(2)}(0)$ with the input sequence $\{u^{(1)}(t)-u^{(2)}(t)\}$ produces the zero output sequence which contradicts the property given in Corollary 5. The necessity can be concluded using the similar argument. For the sake of contradiction let us assume this property does not hold, i.e., there exists a non zero initial state $x(0)\neq 0$ that corresponds to the zero output sequence. This contradicts the strong observability since the zero output sequence can be generated from both zero and $x(0)\neq 0$ as initial conditions under (possibly different) input sequences.

PROOF. [Lemma 13] We prove this lemma with contradiction. We show that if TEST($\Gamma_u^{\text{cert}},\Gamma_y^{\text{temp}}\cup\{i\}$) returns true for all $i\in\Gamma_y^{\text{SAT}}\setminus\Gamma_y^{\text{temp}}$ then TEST($\Gamma_u^{\text{cert}},\Gamma_y^{\text{SAT}}$) would also return true, which contradicts the assumption of the lemma. By applying the following lemma successively, the result follows directly.

Lemma 16 *Assume that the system S is $(2r,2s)$-sparse strongly observable. Pick any subset of inputs and outputs denoted by Γ_u^{cert} and Γ_y^{temp} with $|\Gamma_u^{cert}|\leq 2r$ and $|\Gamma_y^{temp}|\geq p-2s$. Then for any subsets of outputs denoted by Γ_y^1 and Γ_y^2, the first statement implies the second:*

(1) TEST($\Gamma_u^{cert},\Gamma_y^{temp}\cup\Gamma_y^1$) and TEST($\Gamma_u^{cert},\Gamma_y^{temp}\cup\Gamma_y^2$) return true.
(2) TEST($\Gamma_u^{cert},\Gamma_y^{temp}\cup\Gamma_y^1\cup\Gamma_y^2$) returns true.

PROOF. Without loss and generality we can assume Γ_y^1,Γ_y^2 and Γ_y^{temp} are all disjoint sets. Since TEST($\Gamma_u^{\text{cert}},\Gamma_y^{\text{temp}}\cup\Gamma_y^i$) returns true for $i\in\{1,2\}$, therefore we have

$$\begin{bmatrix}\mathbf{Y}|_{\Gamma_y^{\text{temp}}}\\ \mathbf{Y}|_{\Gamma_y^1}\end{bmatrix}=\begin{bmatrix}\mathscr{O}_{\Gamma_y^{\text{temp}}}\\ \mathscr{O}_{\Gamma_y^1}\end{bmatrix}\hat{x}^1+\begin{bmatrix}\mathscr{N}_{\Gamma_u^{\text{cert}}\to\Gamma_y^{\text{temp}}}\\ \mathscr{N}_{\Gamma_u^{\text{cert}}\to\Gamma_y^1}\end{bmatrix}\hat{\mathbf{U}}^1,\qquad(.1)$$

$$\begin{bmatrix}\mathbf{Y}|_{\Gamma_y^{\text{temp}}}\\ \mathbf{Y}|_{\Gamma_y^2}\end{bmatrix}=\begin{bmatrix}\mathscr{O}_{\Gamma_y^{\text{temp}}}\\ \mathscr{O}_{\Gamma_y^2}\end{bmatrix}\hat{x}^2+\begin{bmatrix}\mathscr{N}_{\Gamma_u^{\text{cert}}\to\Gamma_y^{\text{temp}}}\\ \mathscr{N}_{\Gamma_u^{\text{cert}}\to\Gamma_y^2}\end{bmatrix}\hat{\mathbf{U}}^2,\qquad(.2)$$

where $\hat{x}^1,\hat{x}^2\in\mathbb{R}^n$ are states that T-solver.check returns, $\hat{\mathbf{U}}^1,\hat{\mathbf{U}}^2$ are matrices with appropriate dimensions that satisfy TEST. Note that the underlying system is $(2r,2s)$-sparse strongly observable, $|\Gamma_u^{\text{cert}}|\leq 2r$ and $|\Gamma_y^{\text{temp}}|\geq p-s$ therefore $\hat{S}:=(A,B_{(.,\Gamma_u^{\text{cert}})},C_{(\Gamma_y^{\text{temp}},.)},D_{(\Gamma_y^{\text{temp}},\Gamma_u^{\text{cert}})})$ is strongly observable. One can reinterpret $(\hat{\mathbf{U}}^1,\mathbf{Y}|_{\Gamma_y^{\text{temp}}})$ and $(\hat{\mathbf{U}}^2,\mathbf{Y}|_{\Gamma_y^{\text{temp}}})$ as two (possibly different) valid trajectories of a strongly observable system $\hat{S}$ with identical output sequences. Strong observability implies that the state can be uniquely determined from the output with a delay bounded by n, therefore $\hat{x}^1=\hat{x}^2$. Furthermore, the equality of right hand sides of (.1) and (.2) implies that,

$$\mathscr{N}_{\Gamma_u^{\text{cert}}\to\Gamma_y^{\text{temp}}}(\hat{\mathbf{U}}^2-\hat{\mathbf{U}}^1)=0,\qquad(.3)$$

i.e., $\hat{\mathbf{U}}^2 - \hat{\mathbf{U}}^1$ is a zero dynamic of $\hat{S}$. By $(2r, 2s)$-sparse strongly observablity of S, we conclude that $\hat{\mathbf{U}}^2 - \hat{\mathbf{U}}^1$ is also a zero dynamic of S, and therefore,

$$\mathscr{N}_{\Gamma_u^{\text{cert}} \to \Gamma_y^1}(\hat{\mathbf{U}}^2 - \hat{\mathbf{U}}^1) = 0, \quad \mathscr{N}_{\Gamma_u^{\text{cert}} \to \Gamma_y^2}(\hat{\mathbf{U}}^2 - \hat{\mathbf{U}}^1) = 0. \quad (.4)$$

Putting (.1), (.2) and (.4) together with $\hat{x}^1 = \hat{x}^2$, we conclude that:

$$\begin{bmatrix} \mathbf{Y}|_{\Gamma_y^{\text{temp}}} \\ \mathbf{Y}|_{\Gamma_y^1} \\ \mathbf{Y}|_{\Gamma_y^2} \end{bmatrix} = \begin{bmatrix} \mathscr{O}_{\Gamma_y^{\text{temp}}} \\ \mathscr{O}_{\Gamma_y^1} \\ \mathscr{O}_{\Gamma_y^2} \end{bmatrix} \hat{x}^1 + \begin{bmatrix} \mathscr{N}_{\Gamma_u^{\text{cert}} \to \Gamma_y^{\text{temp}}} \\ \mathscr{N}_{\Gamma_u^{\text{cert}} \to \Gamma_y^1} \\ \mathscr{N}_{\Gamma_u^{\text{cert}} \to \Gamma_y^2} \end{bmatrix} \hat{\mathbf{U}}^1, \quad (.5)$$

i.e., TEST($\Gamma_u^{\text{cert}}, \Gamma_y^{\text{temp}} \cup \Gamma_y^1 \cup \Gamma_y^2$) returns false.

PROOF. [Proof Sketch of Lemma 15] Let us revisit the optimization (20) inside the consistency check TEST(Γ_u, Γ_y),

$$\min_{\hat{x}, \hat{\mathbf{U}}} \left\| \mathbf{Y}|_{\Gamma_y} - \begin{bmatrix} \mathscr{O}_{\Gamma_y}, \mathscr{N}_{\Gamma_u \to \Gamma_y} \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{\mathbf{U}} \end{bmatrix} \right\| \quad (.6)$$

For a generic LTI system, the matrix $\begin{bmatrix} \mathscr{O}_{\Gamma_y}, \mathscr{N}_{\Gamma_u \to \Gamma_y} \end{bmatrix} \in \mathbb{R}^{n|\Gamma_y| \times n(1+|\Gamma_u|)}$ is of full rank, where n is the order of the LTI system. If $\begin{bmatrix} \mathscr{O}_{\Gamma_y}, \mathscr{N}_{\Gamma_u \to \Gamma_y} \end{bmatrix} \in \mathbb{R}^{n|\Gamma_y| \times n(1+|\Gamma_u|)}$ is of full row rank, then TEST(Γ_u, Γ_y) is satisfied irrespectively of the actual values of $\mathbf{Y}|_{\Gamma_y}$. Therefore in order to have a certificate constructed by inputs in $\overline{\Gamma}_u$ and outputs in Γ_y, $\begin{bmatrix} \mathscr{O}_{\Gamma_y}, \mathscr{N}_{\Gamma_u \to \Gamma_y} \end{bmatrix} \in \mathbb{R}^{n|\Gamma_y| \times n(1+|\Gamma_u|)}$ should be a full column rank matrix, therefore

$$n|\Gamma_y| \geq n(1 + |\Gamma_u|). \quad (.7)$$

The certificate consists of inputs in $\overline{\Gamma}_u$ and outputs in Γ_y, therefore the length of certificate is:

$$|\overline{\Gamma}_u| + |\Gamma_y| = m - |\Gamma_u| + |\Gamma_y| \geq m + 1. \quad (.8)$$