A Survey of Physics-Based Attack Detection in Cyber-Physical Systems

JAIRO GIRALDO, DAVID URBINA, ALVARO CARDENAS, JUNIA VALENTE, MUSTAFA FAISAL, and JUSTIN RUTHS, University of Texas at Dallas
NILS OLE TIPPENHAUER, Singapore University of Technology and Design
HENRIK SANDBERG, KTH Royal Institute of Technology
RICHARD CANDELL, National Institute of Standards and Technology

76

Monitoring the "physics" of cyber-physical systems to detect attacks is a growing area of research. In its basic form, a security monitor creates time-series models of sensor readings for an industrial control system and identifies anomalies in these measurements to identify potentially false control commands or false sensor readings. In this article, we review previous work on physics-based anomaly detection based on a unified taxonomy that allows us to identify limitations and unexplored challenges and to propose new solutions.

CCS Concepts: • **Security and privacy** → **Intrusion detection systems**; **Spoofing attacks**; • **Information systems** → *Process control systems*; • **Computing methodologies** → *Control methods*;

Additional Key Words and Phrases: Cyber-physical systems, metrics, industrial control systems

ACM Reference format:
Jairo Giraldo, David Urbina, Alvaro Cardenas, Junia Valente, Mustafa Faisal, Justin Ruths, Nils Ole Tippenhauer, Henrik Sandberg, and Richard Candell. 2018. A Survey of Physics-Based Attack Detection in Cyber-Physical Systems. *ACM Comput. Surv.* 51, 4, Article 76 (July 2018), 36 pages.
https://doi.org/10.1145/3203245

1 INTRODUCTION

One of the fundamentally unique properties of industrial control—when compared to general Information Technology (IT) systems—is that the physical evolution of the state of a system has to follow immutable laws of nature. For example, the physical properties of water systems (fluid dynamics) or the power grid (electromagnetics) can be used to create time-series models that we can

The work at UT Dallas was supported by NIST Awards No. 70NANB16H019 and No. 70NANB17H282 and by NSF Grants No. CNS-1553683 and No. CNS-1718848. The work at SUTD was supported by the NRF Singapore (Grant No. NRF2014NCR-NCR001-40). H. Sandberg was supported in part by the Swedish Research Council (Grant No. 2013-5523) and the Swedish Civil Contingencies Agency through the CERCES project.
Authors' addresses: J. Giraldo, D. Urbina, A. Cardenas, J. Valente, M. Faisal, and J. Ruths are with the Erik Jonsson School of Engineering & Computer Science at the University of Texas at Dallas, 800 W. Campbell Rd., Richardson, USA 75080; emails: {jairo.giraldo, david.urbina, junia.valente, mustafa.faisal, jruths}@utdallas.edu; N. O. Tippenhauer is with the Information Systems Technology and Design Pillar at the Singapore University of Technology and Design, 8 Somapah Rd., Singapore 487372; email: nils_tippenhauer@sutd.edu.sg; H. Sandberg is with the Department of Automatic Control at the KTH Royal Institute of Technology, Brinellvägen 8, Stockholm, Sweden 114 28; email: hsan@kth.se; R. Candell is with the Networked Control Systems Group at the National Institute of Standards and Technology, 100 Bureau Dr., Gaithersburg, USA 20899; email: richard.candell@nist.gov.

https://doi.org/10.1145/3203245

then use to confirm that the control commands sent to the field were executed correctly and that the information coming from sensors is consistent with the expected behavior of the system. For example, if we open an intake valve, we should expect that the water level in the tank should rise, otherwise, we may have a problem with the control, actuator, or the sensor; this anomaly can be either due to an attack or a faulty device.

The idea of creating models of the normal operation of control systems to detect attacks has been presented in an increasing number of publications appearing in security conferences in the past couple of years. Applications include water control systems [31], state estimation in the power grid [60, 61], boilers in power plants [111], chemical process control [13], capturing the physics of active sensors [93], electricity consumption data from smart meters [66], video feeds from cameras [19], medical devices [35], and other control systems [67].

The growing number of publications in the past couple of years clearly shows the importance of leveraging the physical properties of control systems for security; however, we have found that most of the papers focusing on this topic are presented independently, with little context to related work. Therefore, research results are presented with different models, different evaluation metrics, and different experimental scenarios. This disjoint presentation of ideas is a limitation for creating the foundations necessary for discussing results in this field and for evaluating new proposals.

Our contributions include: (i) a systematic survey of this emerging field, presented in a unified way and using a new taxonomy based on three main aspects: (1) attack detection, (2) attack location, and (3) validation. Each aspect can be divided in subcategories that encompass different properties of physics-based anomaly detection. The survey includes papers from fields that do not usually interact, such as control theory journals, information security conferences, and power system journals. We identify the relationships and trends in these fields to facilitate interactions among researchers of different disciplines.

(ii) Based on our review of the work from different domains, we present an analysis of the implicit assumptions made in papers and the trust placed on embedded devices, and a logical detection architecture that can be used to elucidate hidden assumptions, limitations, and possible improvements to each work.

(iii) We show that the status quo for evaluating anomaly detection proposals is not consistent and cannot be used to build a research community in this field. We identify limitations in previous evaluations, and we introduce a metric recently proposed that overcome some of those limitations.

The remainder of this work is organized as follows: The scope of this work is presented in Section 1.1. In Section 2, we provide a brief introduction to control systems, and in Section 3, we present the taxonomy we use to classify related work. We apply our taxonomy to related work in Section 4. In Section 5, we summarize our findings from related work, and point out common shortcomings in the literature. We propose improvements and new research efforts in Sections 6.1 and 6.2. We summarize other surveys in Section 7 and conclude our discussions in Section 8.

1.1 Scope of Our Study

There is a growing literature on the security of Cyber-Physical Systems (CPS), including the verification of control code by an embedded system before it reaches the Programmable Logic Controller (PLC), Remote Terminal Unit (RTU), or Intelligent Electronic Device (IED) [69], security of embedded devices [55], the automatic generation of malicious PLC payloads [68], security of medical devices [86], vulnerability analysis of vehicles [15, 42, 49], and of automated meter readings [1, 85]. There is also ongoing research on CPS privacy including smart grids [43], vehicular location monitoring [37], and location privacy [91]. We consider those works related, but complementary to our work.

This article focuses on the problem of using real-time measurements of the physical world to build indicators of attacks. Our work is motivated by false sensor measurements [60, 98] or false control signals like manipulating vehicle platoons [27], manipulating demand-response systems [98], and the sabotage Stuxnet [25, 54] created by manipulating the rotation frequency of centrifuges. The question we address is how to detect these false sensor or false control attacks.

One of the first papers to consider intrusion detection in industrial control networks was Cheung et al. [16]. Their work articulated that network anomaly detection might be more effective in control networks where communication patterns are more regular and stable than in traditional IT networks. Similar work has been done in smart grid networks [1, 9] and in general CPS systems [72]; however, as Hadvziosmanovic et al. showed [30], intrusion detection systems that fail to incorporate domain-specific knowledge and the context in which they are operating will still perform poorly in practical scenarios. Even worse, an attacker that has obtained control of a sensor, an actuator, or a PLC can send manipulated sensor or control values to the physical process while complying to typical traffic patterns such as Internet Protocol (IP) addresses, protocol specifications with finite automata or Markov models, connection logs, and so on.

In contrast to work in CPS intrusion detection that focuses on monitoring such low-level IT observations, in this article, we systematize the recent and growing literature in computer security conferences (e.g., CCS'15 [93], CCS'09 [60], ACSAC'13 [67], ACSAC'14 [31], ASIACCS'11 [13], and ESORICS'14 [111]) studying how monitoring sensor values from physical observations, and control signals sent to actuators, can be used to detect attacks. We also systematize similar results by other fields like control theory conferences with the goal of helping security practitioners understand recent results from control theory, and control theory practitioners understand research results from the security community. Our selection criteria for including a paper in the survey is to identify all the papers (that we are aware of) where the system monitors sensor and/or control signals, and then raises an alert whenever these observations deviate from a model of the physical system.

2 BACKGROUND

We now briefly introduce control systems, common attacks, and countermeasures proposed in the literature.

2.1 Background

A general feedback control system has four components: (1) the physical phenomena of interest (sometimes called the "plant"), (2) sensors to observe the physical system and send a time series y_k denoting the value of the physical measurement at time k (e.g., the voltage at 3am is 120KV), (3) based on the sensor measurements received y_k, the controller sends control commands u_k (e.g., open a valve by 10%) to actuators, and (4) actuators that change the control command to an actual physical change (the device that opens the valve).

A general security monitoring architecture for control systems that looks into the "physics" of the system needs an anomaly detection system that receives as inputs the sensor measurements y_k from the physical system and the control commands u_k sent to the physical system and then uses them to identify any suspicious sensor or control commands is shown in Figure 1.

The idea of monitoring sensor measurements y_k and control commands u_k and to use them to identify problems with sensors, actuators, or controllers is not new. In fact, this is what the literature of fault-detection in dynamical systems has investigated for more than four decades [28, 41, 115]. Fault Detection, Isolation, and Reconfiguration (FDIR) methods are diverse and encompass research on hardware redundancy (e.g., adding more sensors to detect faulty measurements, or adding more controllers and decide on a majority voting control) as well as software (also known as analytical) redundancy [41]. While fault-detection theory provides the foundations for our work, the disadvantage of fault-detection systems is that they were designed to detect and respond to equipment failures, random faults, and accidents, not attacks.

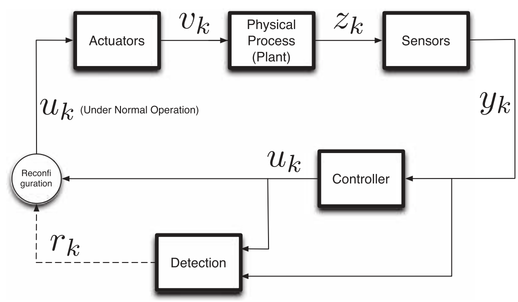

Fig. 1. Anomaly detection architecture. The sensor measurements y_k and the control commands u_k are fed to the anomaly detection block. Under normal operating conditions, the actuation on the plant corresponds to the intended action by the controller: $v_k = u_k$, and the observations are correctly reported back to the controller: $y_k = z_k$.

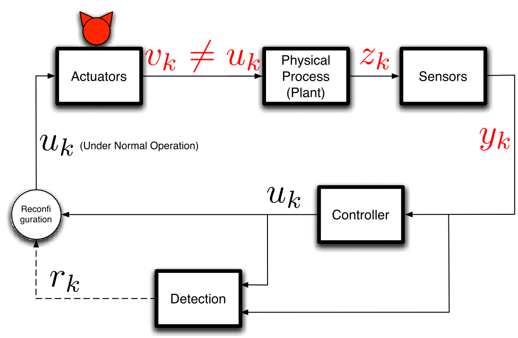

Fig. 2. When one or more actuation signals are compromised (e.g., the actuator itself is compromised or it receives and accepts a control command from an untrusted entity), the actuation to the plant will be different to the intended action by the controller: $v_k \neq u_k$. This false actuation will in turn affect the measured variables of the plant z_k, which in turn affect the sensor measurements reported back to the controller: $y_k = z_k$.

Figure 2 shows an attack on the actuator, which modifies the control command send to the plant. Note that the controller is not aware of the communication interruption. However, Figure 3 shows an attack in the sensor, which allows the attacker to deceive the controller about the real state of the plant. In the worst case, the control device can be compromised as well, giving the attacker potentially unlimited control on the plant to implement any outcome (see Figure 4). This last figure also captures the threat model from a malicious control command sent from the control center as seen in Figure 5: While the implementation might be different—one monitor is placed in the supervisory network and the other monitor on the field communications interface—the logical architecture—what the monitoring application sees—will be the same. In these attack schemes, we assume that the control has a trusted detection mechanism, which can recognize unexpected behaviors and potentially take counter measures.

The detection block in Figures 1–4 is expanded in Figure 6 to illustrate several alternative algorithms we found in the literature. There are two blocks that are straightforward to implement: (1) The *controller* block in Figure 6 is a redundant control algorithm (i.e., in addition to the controller of Figure 1) that checks if the controller is sending the appropriate u_k to the field, and (2) The *safety check* block is an algorithm that checks if the predicted future state of the system will violate a safety specification (e.g., the pressure in a tank will exceed its safety limit). The

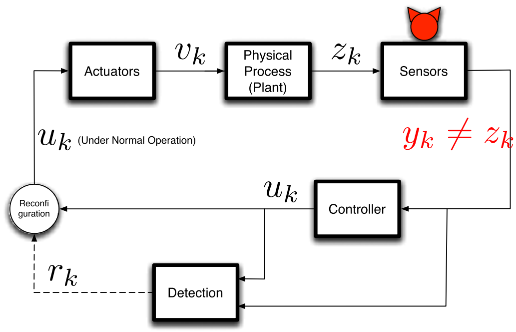

Fig. 3. When one or more sensor signals are compromised (e.g., the sensor itself is compromised or the controller receives and accepts a sensor measurement from an untrusted entity), the sensor measurement used as an input to the control algorithm will be different from the real state of the measured variables $y_k \neq z_k$.

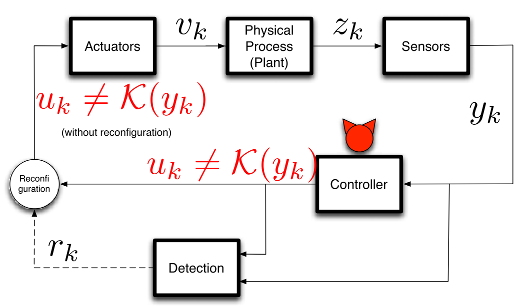

Fig. 4. When the controller is compromised, it will generate a control signal that does not satisfy the logic of the correct control algorithm: $u_k \neq \mathcal{K}(y_k)$.

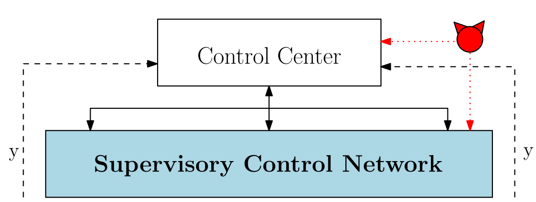

Fig. 5. Attacks on Central Control or Supervisory Control Network translate on the logical model shown in Figure 4.

different alternative detection algorithms are also summarized in Table 1. In this article, we focus on analyzing the more challenging algorithms:

(1) *Prediction* (Physical Model): given sensor y_k and control commands u_k, a model of the physical system will predict a future expected measurement $\hat{y}_{k+1}$.
If we only have output data (sensor measurements y_k), then regression models like AR, ARMA, or ARIMA are a popular way to learn the correlation between observations. Using these models, we can predict the next outcome. For example, for an Auto-Regressive (AR) model, the prediction would be

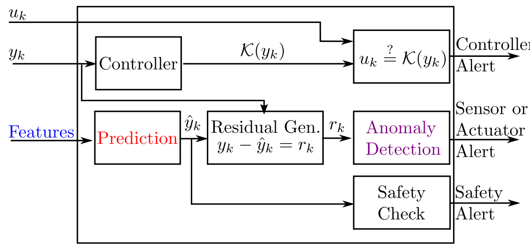

Fig. 6. The detection block from Figure 1, with a set of different detection algorithms. In the top, the *controller* block is a redundant control (i.e., in addition to the controller of Figure 1) that checks if the control commands are appropriate. The middle row (*prediction*, *residual generation*, and *anomaly detection* blocks) focuses on looking at the sensor values and raising an alarm if they are different than what we expect/predict. The *prediction* and *safety check* blocks focus on predicting the future state of the system, and if it violates a safety limit, then we raise an alert.

Table 1. Detection Algorithm Alternatives Found in Literature

Features	
Cur. In & Prev. Out	u_k, y_{k-1}
Prev. Sensor Observ.	$y_{k-1}, y_{k-2}, \ldots, y_{k-N}$
Prediction	
Input-Output LDS	$x_{k+1} = Ax_k + Bu_k v + \epsilon_k$ $y_k = Cx_k + Du_k + e_k$
Output-Only AR	$y_{k+1} = \sum_{i=k-N}^{k} \alpha_i y_i + \alpha_0 + \epsilon_k$
Anomaly Detection	
Stateless	$\lvert r_k \rvert \stackrel{?}{>} \tau$
Stateful	$S_0 = 0.\ \ (S_k + \lvert r_k \rvert - \delta)^+ \stackrel{?}{>} \tau$

$$\hat{y}_{k+1} = \sum_{i=k-N}^{k} \alpha_i y_i + \alpha_0, \tag{1}$$

where α_i are the coefficients learned through system identification and y_i the last N sensor measurements—where the amount of parameters to learn N can be also estimated to prevent over-fitting of the model using tools like Akaike's Information Criteria (AIC). It is possible to obtain the coefficients α_i, by solving an optimization problem that minimizes the residuals (e.g., least squares) [62].

If we have inputs (control commands u_k) and outputs (sensor measurements y_k) available, then we can use *subspace model identification* methods, producing the following model:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + \epsilon_k, \\ y_k &= Cx_k + Du_k + e_k, \end{aligned} \tag{2}$$

where A, B, C, and D are matrices modeling the dynamics of the physical system. Most physical systems are strictly causal and then therefore usually $D = 0$. The control commands $u_k \in \mathcal{R}^p$ affect the next time step of the state of the system $x_k \in \mathcal{R}^n$ and sensor measurements $y_k \in \mathcal{R}^q$ are modeled as a linear combination of these hidden states. e_k and ϵ_k are sensor and perturbation noise, and are assumed to be a random process with zero mean.

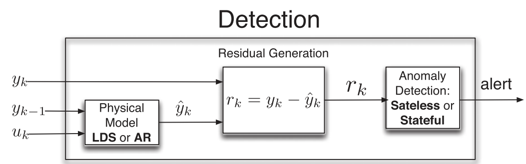

Fig. 7. The detetion module from Figure 6 focusing on using anomaly detection based on the physics of the process.

(2) *Anomaly detection* (Statistical Test): Given a time series of residuals r_k (the difference between the received sensor measurement y_k and the predicted/expected measurement $\hat{y}_k$), the anomaly detection test needs to determine when to raise an alarm. Anomaly detection strategies that are based on residuals, can be divided into two main categories: Statelss and Stateful.
In a **Stateless** test, we raise an alarm for every single significant deviation at time k, i.e., if $|y_k - \hat{y}_k| = r_k \geq \tau$, where τ is a threshold.
In a **Stateful** test, we compute an additional statistic S_k that keeps track of the historical changes of r_k (no matter how small) and generate an alert if $S_k \geq \tau$, i.e., if there is a persistent deviation across multiple time steps. There are many tests that can keep track of the historical behavior of the residual r_k such as taking an average over a time-window, an exponential weighted moving average (EWMA), or using change detection statistics such as the non-parametric CUmulative SUM (CUSUM) statistic.
The theory behind CUSUM assumes we have a probability model for our observations r_k (the residuals in our case); this obscures the intuition behind CUSUM, so we focus on the non-parametric CUSUM (CUSUM without probability likelihood models), which is basically a sum of the residuals. In this case, the CUSUM statistic is defined recursively as $S_0 = 0$ and $S_{k+1} = (S_k + |r_k| - \delta)^+$, where $(x)^+$ represents $\max(0, x)$ and δ is selected so that the expected value of $|r_k| - \delta < 0$ under hypothesis H_0 (i.e., δ prevents S_k from increasing consistently under normal operation). An alert is generated whenever the statistic is greater than a previously defined threshold $S_k > \tau$ and the test is restarted with $S_{k+1} = 0$.

By focusing on these algorithms our detection block can be simplified as shown in Figure 7.

2.2 State Estimation

Before we start our survey, we also need some preliminaries in what state estimation is. Whenever the sensor measurements y_k do not observe all the variables of interest from the physical process, we can use state estimation to obtain an estimate $\hat{x}_k$ of the real state of the system x_k at time k (if we have a model of the system).

Recall Equation (2) gives us the relationship between the observed sensor measurements y_k and the hidden state x_k. The naive approach would assume the noise e_k is zero and then solve for x_k: $x_k = C^{-1}(y_k - Du_k)$; however, for most practical cases this is not possible as the matrix C is not invertible, and we need to account for the variance of the noise. The exact solution for this case goes beyond the scope of this article, but readers interested in finding out how to estimate the state of a dynamical system are encouraged to read about Luenberger observers [96] and the Kalman filter [112], which are used to dynamically estimate the system's states without or with noise, respectively.

State estimates can then be used for the control logic, for prediction (and therefore for bad data detection), and for safety checks, as in Figure 8.

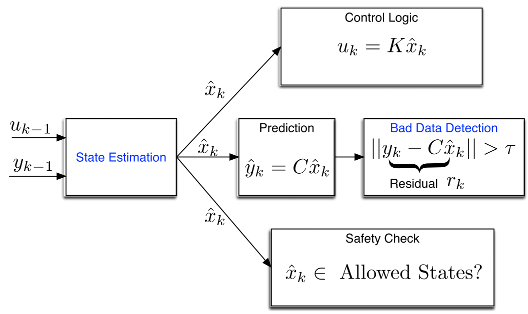

Fig. 8. Whenever the sensor measurements y_k do not observe all the variables of interest from the physical process, we can use state estimation to obtain an estimate $\hat{x}_k$ of the real state of the system x_k at time k (if we have a model of the system). State estimates can then be used for the control logic, for prediction (and therefore for bad data detection), and for safety checks.

In addition to the literature on state estimation in the power grid [61], there has been work studying the role of state estimation for the security of other cyber-physical systems. For example, Chong et al. introduce secure state estimation for a generic control system described in Equation (2) where stateful detection strategies are used to search for a subset of sensors that are not under attack to generate accurate estimations. This approach started a novel line of research that helps to mitigate the impact of attacks for those cases where estimation is a fundamental part of the control and decision making (e.g., observer based control).

3 TAXONOMY

We now present our new taxonomy, which attempts to identify the key characteristics for the literature on physics-based attack-detection. Figure 9 illustrates our taxonomy, and Table 2 summarizes the application of our taxonomy to the related literature (done in the next section). In particular, our taxonomy has the following characteristics:

(1) **Attack Detection**: The methods proposed by researchers to detect attacks. Attack detection is divided in two:
   (a) **Prediction**: The model of used by the researchers to predict the state of the system.
   (b) **Detection Statistic**: How the anomaly is scored, and the conditions for raising an alert.
(2) **Attack Location**: The specific device that launches the attack; e.g., the sensor, controller, or actuator.
(3) **Validation**: How researchers validate their attack-detection algorithms. This is further divided in two points:
   (a) **Metrics:** How researchers measure the effectiveness of the detection algorithm.
   (b) **Implementation:** How the attack-detection algorithms are implemented, e.g., researchers can do simulations, use real-world systems, or obtain data from operators.

3.1 Attack Detection

As we mention above, physics-based anomaly detection mechanisms rely on an adequate prediction of the system behavior. We now describe the options for modeling the physical system and the ways to determine whether or not to raise an alarm.

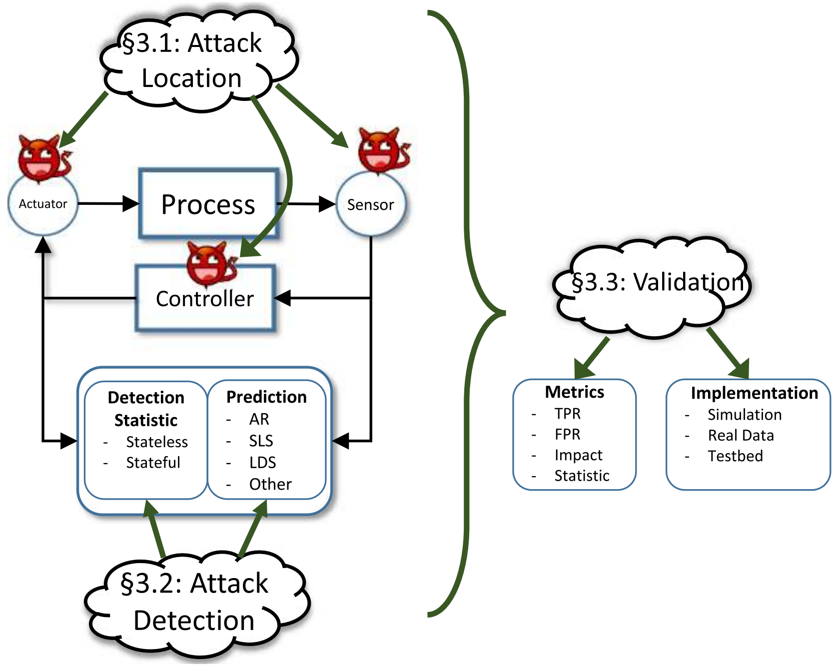

Fig. 9. Our proposed taxonomy focuses on three components: Attack Detection (Section 3.1), Attack Location (Section 3.2), and Validation (Section 3.3).

3.1.1 Prediction Model: LDS or AR. The model of how a physical system behaves can be developed from physical equations (Newton's laws, fluid dynamics, or electromagnetic laws) or it can be learned from observations through a technique called *system identification* [5, 63]. In system identification one often has to use either Auto-Regressive Moving Average with eXogenous inputs (ARMAX) or linear state-space models. Two popular models used by the papers we survey are **Auto-Regressive (AR)** models (e.g., used by Hadziosmanovic et al. [31]) and **Linear Dynamical State-Space (LDS)** models (e.g., used by PyCRA [93]). AR models are a subset of ARMAX models but without modeling external inputs or the average error and LDS are a subset of state space models.

To make a prediction, we need y_k (and u_k for LDS) to obtain a $\hat{y}_{k+1}$. For the particular case of LDS, a *state estimator* uses y_k, u_k to obtain $\hat{x}_{k+1}$, and then compute $\hat{y}_{k+1} = C\hat{x}_{k+1}$ (if D is not zero, then we also need u_{k+1}). Some communities adopt models that employ the observation equation from Equation (2) without the dynamic state equation. In this particular case, it is possible to assume that the at each sampling instant, the system has already reached some (quasi)stable state. We refer to this special case of LDS as **Static Linear State-space (SLS)** model.

3.1.2 Detection Statistic: Stateless or Stateful. Based on the observed sensor or control signals up to time k, we can use models of the physical system (e.g., AR or LDS) to predict the expected observations $\hat{y}_{k+1}$ (note that $\hat{y}_{k+1}$ can be a vector representing multiple sensors at time $k+1$). The difference r_k between the observations predicted by our model $\hat{y}_{k+1}$ and the sensor measurements received from the field y_{k+1} is usually called a **residual**. If the observations we get from the sensors y_k are significantly different from the ones we expect (i.e., if the residual is large), then we can generate an alert. To measure that difference, we identify two types of detection statistics: *stateless*, which are evaluated at each time instant, or stateful, which keep track of historical changes.

3.2 Attack Location

To evaluate attack detection schemes, it is important to explicitly state which components in the control loop need to be trusted to correctly detect attacks. We found in the literature that sometimes detection mechanisms are proposed without a clear definition of what the attacker can do and cannot do. In particular, one of the discussions we have later in the article is that attacks can bypass attack-detection methods depending on the specific location from where they are launched. In particular, we focus on attackers that can compromise sensors, actuators, or controllers.

3.3 Validation

We classify the way attack-detection proposals are validated in two main groups: (i) implementation and (ii) evaluation metrics.

3.3.1 Implementation. The proposed detection techniques are typically implemented using different types of experimental scenarios to evaluate their efficacy and illustrate how the physical system is affected by cyber attacks. We found in the literature three types of experimental settings: (1) simulations, (2) data from real-world systems (e.g., data captured from a power utility), and (3) real-world testbeds (real-world systems that are not operational, but are used solely for experimentation). Each of them possesses some advantages but also some drawbacks. For instance, with simulations, it is possible to test the impact of attacks without any safety hazard, but doing so ignores many of the real-world problems practitioners encounter. However, deploying attacks in a testbed enables a more realistic analysis of vulnerabilities of the system, but it comes with the risks of damaging expensive equipment or even harming people.

3.3.2 Evaluation Metrics. The evaluation metric is used to determine the effectiveness of the proposed detection scheme. Ideally, the metric should allow for a fair comparison of different schemes that are targeting the same adversarial model for comparable settings. Common evaluation metrics are the number of false alerts and the probability of detecting attacks. A parametric curve illustrating the trade-off of these two quantities is the Receiver Operating Characteristic (ROC) curve. A specific combination of these two metrics into a single quantity is the *accuracy* (correct classification) of the anomaly detector.

We also found in the literature other evaluation metrics, such as the impact that the attack is able to cause in the physical process (e.g., deviation from an operation point), the evolution of the detection statistic over time, and the *time of detection*, which quantifies how long it takes to detect an attack right after it is launched. This latter metric is very important, because the real-time safety-criticality of most control systems. It does not matter if an attack-detection algorithm can detect an attack if by the time it is detected

In the next section, we apply our taxonomy to a survey of the literature on physics-based attack detection for CPS.

4 SURVEY OF PREVIOUS WORK

The term Cyber-Physical Systems (CPS) was coined over a decade ago as an attempt to unify the emerging application of embedded computer and communication technologies to a variety of physical domains, including aerospace, automotive, chemical production, civil infrastructure, energy, healthcare, manufacturing, materials, and transportation. The goal of CPS research is to reveal cross-cutting fundamental scientific and engineering principles that underpin the integration of cyber and physical elements across all application sectors. As such, research communities from established different backgrounds ranging from *control theory*, *power systems*, and *cyber-security* have tried to provide their own solutions to physics-based attack detection.

In the cross-cutting spirit of CPS, we divide our survey based on how physics-based attack detection is approached by different disciplines: (1) control theory (articles published in control theory conferences, or journals), (2) power systems (articles published in power systems conferences, or journals), and (3) cyber-security (articles published in security conferences, or journals). Our goal is to identify what are the trends and common points when researchers from one field (e.g., control theory) study CPS security and then see how it compares with other fields (e.g., cyber-security).

We assigned workshops to the venue that the main conference is associated with. We also assigned conferences associated with Cyber-Physical Systems Week (CPSWeek) to control conferences because of the overlap of attendees to CPSWeek coming with control theory background.

4.1 Control Theory

There is a significant body of work in attack detection from the control theory community [7, 8, 38, 53, 70], which does not focus on a specific domain. While the treatment of the topic is highly mathematical (a recent special issue of the IEEE Control Systems Magazine provides an accessible introduction to the topic [39]), we attempt to extract the intuition behind key approaches to see if they can be useful for the computer security community.

Most control theory papers we reviewed look at *models of the physical system* satisfying Equation (2), because that model has proven to be very powerful for most practical purposes. In addition, several of these papers assumed a stateless *detection*. We think this bias towards the stateless test by the control theory community stems from the fact that the stateless test allows researchers to prove theorems and derive clean mathematical results. In contrast, providing such thorough theoretical analysis for stateful tests (e.g., CUSUM) can become intractable for realistic systems. We believe that this focus on strong analytical results prevents the use of stateful tests that effectively perform better in many practical cases. Most of the stateful approaches that can be found in control theory papers focus on windowed χ^2 [76], and combinatorial optimization approaches [71]. For instance, Murguia et al. [78] have proposed an extensive analysis of the non-parametric CUSUM to guarantee an adequate tuning that depends on the system dynamics and its stochastic properties.

In addition to these trends, we identified several novel ideas from the control theory community; we summarize them below as (1) secure state estimation, (2) zero-dynamics attacks, (3) combined cyber and physical attacks, (4) active monitoring, and (5) energy-based methods.

Secure State Estimation. One of the main areas of research in the control theory community is to find efficiently the subset of sensors that are sending false information [17, 92]. *Attack Detection:* The system models satisfy Equation (2). The main idea behind these papers consists on solving a combinatorial optimization problem to find a subset of sensors whose elements *are not under attack*, to generate adequate state estimations. When multiple sensors are under attack, this problem is shown to be NP-hard (combinatorial in the number of sensors), so the goal of research papers is to find efficient algorithms under a variety of assumptions. Recently, Shoukry et al. [92] proposed a search algorithm based on Satisfiability Modulo Theory (SMT) to speed up the search of possible sensors sets; this research has been extended for systems subject to random noise [71]. *Attack Location*: A subset of sensors can be compromised. In particular, it was found that the total number of sensors used to monitor the process has to be at least twice the number of sensors under attack. From this discussion, we can infer that the authors assumed controller and actuators to be trusted. Another paper considering secure state estimation where the attacks are located in the control signal is done by Yong et al. [120]. *Validation*: The theoretical results are tested using simulations where the accuracy of the estimation and searching times are compared for different searching methods to verify the effectiveness of SMT.

These papers are also among the few that discuss the *time-to-detect* metric and offer solutions trying to minimize it. In our discussion section, we will return to the need to have more papers discussing the time to detect an attack.

Zero-dynamics Attacks. These attacks are interesting, because they show that even without compromising sensors, attackers can mislead control systems into thinking they are at a different state. The attacks require the attacker to compromise the actuators, that the anomaly detection system monitors the legitimate control signal u_k and the legitimate sensor signal y_k, and a plant vulnerable to these attacks.

One of the fundamental properties control engineers ask about Equation (2) is whether or not the system is *Observable* [96]. If it is observable, then we know that we can obtain a good state estimate $\hat{x}_k$ given the history of previous control inputs u_k and sensor measurements y_k. Most practical systems are observable or are designed to be observable. Now, if we assume an observable system, then we can hypothesize that *the only way to fool a system into thinking it is at a false state, is by compromising the sensors and sending false sensor readings.* Zero-dynamics attacks are an example that this hypothesis is false [83, 102, 103].

Zero-dynamics attacks require attackers that inject fake signals in sensors or actuators, and modify "hidden" (or unobservable) states. For instance, in Figure 2 the anomaly detector observes a valid u_k and a valid y_k, but it does not observe the compromised v_k. Not all systems are vulnerable to these attacks, but certain systems like the quadruple tank process [44] can be (depending on the specific parameters).

Though zero-dynamics attacks are interesting from a theoretical point of view, most practical systems will not be vulnerable to these attacks (although it is always good to check these conditions). First, if the sensors monitor all variables of interest, we will not need to state estimation (although this might not be possible in a large-scale control system with thousands of states); second, even if the system is vulnerable to zero-dynamics attacks, the attacker has to follow a specific control action from which it cannot deviate (so the attacker will have problems achieving a particular goal—e.g., move the system to a particular state), and finally, if the system is minimum phase, the attacker might not be able to destabilize the system. In addition, there are several recommendations on how to design a control system to prevent zero-dynamic attacks [103].

Combined Use of Cyber- and Physical Attacks. Control theory papers have also considered the interplay between physical attacks and cyber-attacks. *Attack location:* In a set of papers by Amin et al. [3, 4], the attacker launches physical attacks to the system (physically stealing water from water distribution systems), while at the same time it launches a cyber-attack (compromised sensors send false data masking the effects of the physical attack). We did not consider physical attacks originally, but we then realized that the actuation attacks of Figure 2 account for physical attack, as it is equivalent to the attacker inserting its own actuators, and therefore the real actuation signal v_k will be different from the intended control command u_k. *Attack Detection:* Authors focus on LDS models obtained from an exhaustive analysis of hydrodynamic properties. They propose the use of unknown input observers and stateless detection to identify the location of a possible attacks; however, the bottom line is that if the attackers control enough actuation and sensor measurements, there is nothing the detector can do as the compromised sensors can always send false data to make the detector believe the system is in the state the control wanted it to go. These *covert attacks* have been characterized for linear [94] and nonlinear systems [95]. *Validation:* The results are validated in a real water distribution network plant for different case studies.

GPS Spoofing. Kerns et al. [45] consider how Global Positioning System (GPS) spoofing attacks can take control over unmanned aircrafts. *Attack Location: The spoofer is able to generate a counterfeit GPS signal and sends it to the GPS antenna of the aircraft. The fake signal is designed to replace* *the real GPS reading with a fake position. Attack Detection:* They use an LDS as a model of the physical system, and then use a *stateless* residual (also referred to as innovations) test to detect attacks. Validation: They test different attacks through simulation and also by using an aircraft Hornet Mini UAV, which is a mini helicopter of 4.5Kg. They show two attacks, one where the attacker is detected, and another one where the attacker manages to keep all the residuals below the threshold while still changing the position of the aircraft.

Active Monitoring. Most physics-based anomaly detection algorithms are passive; i.e., they do not affect the system they are observing. In contrast, active monitoring changes the physical system by sending unpredictable control commands and then verifying that the sensor responds as expected. *Attack Detection:* The work of Mo et al. [74–76] considers embedding a watermark in the control signal, by injecting random noise from a known distribution. This is useful for systems that remain constant for long periods of time (if they are in a steady state) and to see if the control action has any effect, the operator needs to send a random signal to the system. They use a time-window stateful anomaly detection statistic. *Attack Location:* This approach can detect sensor or actuator attacks. *Validation:* results are evaluated using simulations and TPR,FPR.

The idea of active monitoring has also been proposed in other domains [75, 93, 106, 107], as we will discuss later in the article.

Energy-based Attack Detection. Finally, another detection mechanism using control theoretic components was proposed by Eyisi and Koutsoukos [24]. *Attack detection:* The main idea is that the energy properties of a physical system can be used to detect errors or attacks. Unlike observer-based detection (used by the majority of control papers), their work uses concepts of energy or passivity, which is a property of systems that consume but do not produce net energy. In other words, the system is passive if it dissipates more energy than it generates. To use this idea for detecting attacks, the monitor function estimates the supplied energy (by control commands) and compares it to the energy dissipated and the energy stored in the system (which depend on the dynamics of the system). While the idea is novel and unique, it is not clear why this approach might be better than traditional residual-based approaches, in particular given that any attack impersonating a passive device would be undetected, and in addition, the designer needs more information. To construct an energy model, a designer needs access to inputs and outputs, the model of the system in state space (as in Equation (2)), and functions that describe the energy dissipation of a system in function of the stored energy (energy function) and the power supply (supply function). *Attack location:* It is assumed that sensors are under attack and that the controller and actuator are trusted, but their results can be easily extended to other attack scenarios. *Validation:* The validation is conducted using simulation results that illustrate the impact of the attack in the system states and how the energy changes can help to indicate whether or not an attack is present.

4.2 Power Systems

The area of power systems has also been the subject of intense study due to the criticality of the power grid. One of the most active areas of research by the power systems community is the problem of bad data detection in state estimation. Interestingly enough, the first paper on this research topic was published in a cyber-security conference [60] and was later extended in a cyber-security journal [61]; however, because this paper motivated several researchers with power systems background to continue this line of research, we are including it along with the other power systems papers.

Attacks on Bad Data Detection. One of the most popular lines of work in the power systems community is the study of false-data injection attacks against state estimation in the power grid.

In the power grid, operators need to estimate the phase angles x_k from the measured power flow y_k in the transmission grid. These bad data detection algorithms were meant to detect random sensor faults, not strategic attacks, and as Liu et al. [60, 61] showed, it is possible for an attacker to create false sensor signals that will not raise an alarm (experimental validation in software used by the energy sector was later confirmed [101]). *Attack Detection:* It is known that the measured power flow $y_k = h(x_k) + e_k$ is a nonlinear noisy measurement of the state of the system x and an unknown quantity e_k called the measurement error. Liu et al. considered the linear model where $y_k = Cx_k + e_k$, therefore this model of the physical system is the sensor measurement SLS model described by Equation (2), where the matrix D is zero and where the dynamic state equation is not specified. The detection mechanism they consider is a stateless anomaly detection test, where the residual is $r_k = y_k - C\hat{x}_k$, the state estimate is defined as $\hat{x}_k = (C^T W^{-1} C)^{-1} C^T W^{-1} y_k$, and W is the covariance matrix of the measurement noise e_k. Note that because r_k is a vector, the metric $|\cdot|$ is a vector distance metric, rather than the absolute value. This test is also illustrated in the middle row of Figure 8. *Attack Location:* The sensor data is manipulated, and cannot be trusted. The goal of the attacker is to create false sensor measurements such that $|r_k| < \tau$. *Validation:* The paper uses a metric that focuses on how hard it is for the adversary to find attacks such that $|r_k| < \tau$. The results are validated using simulation results.

There has been a significant amount of follow up research focusing on false data injection for state estimation in the power grid, including the work of Dán and Sandberg [21], who study the problem of identifying the best k sensors to protect to minimize the impact of attacks (they assume the attacker cannot compromise these sensors). Kosut et al. [50] consider attackers trying to minimize the error introduced in the estimate, and defenders with a new detection algorithm that attempts to detect false data injection attacks. Liang et al. [56] consider the nonlinear observation model $y_k = h(x_k) + e_k$. Further work includes [10, 29, 46, 84, 90, 100, 108].

Automatic Generation Control. While attacks to state estimation can be dangerous, even more catastrophic to the power grid would be an attack against the *Automatic Generation Control* (AGC) signal, as this signal controls the power generation of the bulk power grid. An incorrect AGC signal can cause large transmission systems disconnecting from each other and cause severe blackouts. To compute the AGC signal to be sent to generators, control centers in the power grid send Area Control Error (ACE) signals to ramp up or ramp down generation based on the state of the grid. Sridhar and Govindarasu [97] were one of the first to consider the case when an ACE signal that cannot be trusted. For *Attack Detection*, they use a historical model of how real-time load forecast affects ACE. The ACE computed by the control center (ACE_R) and the one computed from the forecast (ACE_F) are then compared to compute the residual. They add the residuals for a time window and then raise an alarm if it exceeds a threshold. *Attack Location:* The load forecast is trusted but the control ACE signal is not, so the attacker either compromises a machine computing the ACE signal or a control center. For *Validation,* the proposed approach is implemented using simulations, and they use the false positive and false negative detection rates as metrics of performance.

Active Monitoring. Similar to the active monitoring papers we discussed earlier in the article, the works of Morrow et al. [77] and Davis et al. [22] consider *active monitoring*, for power systems. *Attack location:* Adversaries are able to compromise a subset of sensor readings and can design attacks that bypass the anomaly detection strategy. *Anomaly Detection:* The focus is on static linear models of the flow equations of the power grid to generate estimations that are used by an stateless anomaly detection algorithm. In particular, the proposed active monitoring strategy changes randomly the topology of the power grid to increase the effort of an adversary that wants to remain undetected, because this reconfiguration will change the state of the system and if the adversary does not change attack appropriately, then it will be detected. *Validation:* Different benchmarks were simulated to verify the proposed strategy.

While the idea of perturbing the system to reveal attackers that do not adapt to these perturbations is intuitively appealing, it also comes with a big operational cost to power systems: the deviation of a system from an ideal operational state just to test if the sensors have been compromised or not might not sound very appealing to asset owners whose livelihood depends on the optimal operation of a system. However, there is another way to look at this idea: if the control signal u_k is already highly variable (e.g., in the control of frequency generators in the power grid who need to react to constant changes in the power demand of consumers), then the system might already be intrinsically better suited to detect attacks via *passive monitoring*.

4.3 Cyber-Security

The cyber-security community is also very active in the physics-based attack detection space. While the *control theory* and *power systems* communities tend to work at the conceptual level, the cyber-security community has focused on the details and complexities of implementing attack-detection in practice.

Real-world Modbus-based Detection. One of the first papers to analyze network data from a real-world industrial system was the work of Hadziosmanovic et al. [31]. In particular, they showed how to use Modbus (an industrial protocol) traces from a real-world operational system to detect attacks by monitoring the state variables of the system, including: constants, attribute data, and continuous data. We focus on their analysis of continuous data, because this research is a motivation for our own experiments in this article. *Attack Detection:* To model the behavior of continuous sensor observations y_k like the water level in a tank or the water pressure in a pipe, the authors use an AR model as we described in Equation (1). This corresponds to models of individual signals, and as we will show in our experiments, if we can create models that show the correlation of multiple variables, then we can obtain better attack detection algorithms. In fact, that was an observation made by the authors, as they found that multiple variables exhibit similar (even identical) behavior. The detection mechanism raises an alert if (1) the measurement y_k reaches outside of specified limits (this is equivalent to the *Safety Check* box in Figure 6) or (2) y_k produces a deviation in the prediction $\hat{y}_k$ of the autoregressive model (noting that $r_k = y_k - \hat{y}_k$), this is the *stateless* statistical test from Figure 6. *Attack Location:* It is not clear where in the control architecture the real-world data trace was collected. Because deploying a large-scale collection of a variety of devices in a control network is easier at the supervisory control network, it is likely that the real-world traffic monitors data exchanged between the control centers and the PLCs. In this case the PLC must be trusted, and therefore the adversary must attack the actuators or the sensors. *Validation:* The paper focuses on understanding how accurately their AR system models the real-world system and identifying the cases where it fails. They mention that they are more interested in understanding the model fidelity rather than in specific true-/false-alarm rates, and we agree with them, because measuring the true positive rate would be an artificial metric. Understanding the model fidelity is implicitly looking at the potential of false alarms, because deviations between predictions and observations during normal operations are indicators of false alarms. While this is a good approach for the exploratory data analysis done in the paper, it might be misunderstood by future proposals. After all, the rule *never raise an alert* will have zero false alarms (but it will never detect any attack). We discuss this further in Section 5. Authors implement their proposed strategy using real data from a testbed.

State Relation-based Intrusion Detection (SRID). Similar to the work on secure state estimation, SRID [111] attempts to detect attacks and then find the root cause of the attack in an industrial control system. SRID is an outlier in our survey; despite a growing literature that follow similar approaches for the topic of using the physics of CPS to detect attacks, SRID proposes system identification, and bad data detection tests that are unique. *Attack Detection:* Instead of using a traditional and well-understood system identification approach to learn a model of the boiler simulator they study, they propose a set of heuristics they name *feedback correlations* and *forward correlations*; however, we were not able to find a good justification as to why these heuristics are needed, or why they are better than traditional system identification methods. We recommend that for any future work, if the authors propose a new system identification tool (previously untested), they should use a traditional tool to test as a baseline approach. One of the goals of SRID is to identify the location of an attack; but we believe that if we know all the control loops in their boiler simulation, we can create models for each of them and identify the root cause using traditional methods; however, the paper does not mention where other researchers can find the boiler simulator SRID used in the experiments, so we cannot compare our methods to theirs. However, SRID does not specify if they use control and sensor measurements for their anomaly detection, but from the description it appears they use only sensor measurements. SRID proposes a new bad data detection based on alternation vectors, which basically tracks the history of measured variables going up or down. If this time series is not an allowable trend (not previously seen), then the detection test generates an alert. It is not clear why this heuristic can perform better than the traditional residual generation approach. *Attack Location:* The sensors cannot be trusted, but the attacker sends *arbitrary data that falls within the sensor's valid range*. Therefore, this attacker is not strategic and it behaves exactly as random faults. It is not clear therefore how their evaluation will differ whenever there is a sensor fault (within the valid range) or the attacker they propose. *Validation:* SRID measures the successful attack detection rate and the false-alarm rate using simulation results.

Attack-Detection and Response. Cardenas et al. [13] consider a chemical industrial control system. *Attack Detection:* The authors approximate the nonlinear dynamics of the chemical system with an input/output linear system, as we defined in Equation (2). Therefore this model captures the correlations among multiple different observations y_k (with the matrix C) but also the correlation between input u_k and output y_k and is therefore a model that can match the fidelity of observations very closely. The authors use the linear system to predict $\hat{y}_k$ given the previous input u_{k-1} and the previous measurement y_{k-1} and then test whether or not the prediction is close to the observed measurement $r_k = y_k - \hat{y}_k$. They raise an alert if the CUSUM statistic (the stateful test of Figure 6) is higher than a threshold. *Attack Location:* One or more sensors are compromised, and cannot be trusted. The goal of the adversary is to violate the safety of the system: i.e., an attacker that wants to raise the pressure level in the tank above 3,000kPa and at the same time remain undetected by the test. The actuators and the control logic are assumed to be trusted. *Validation:* The paper proposes a control reconfiguration whenever an attack is detected, in particular a switch to open-loop control, meaning that the control algorithm will ignore sensor measurements and will attempt to estimate the state of the system based only on the expected consequences of its own control commands. As a result, instead of measuring the false-alarm rate, the authors measure the impact of a reconfiguration triggered by a false alarm on the safety of the system—in other words, a false alarm must never drive the system to an unsafe state (a pressure inside the tank greater than 3,000kPa). To evaluate the security of the detection algorithm, the authors also test to see if an attacker that wants to remain undetected can drive the pressure inside the tank above 3,000kPa. All the results were tested via simulation.

Clustering. Another approach to detect attacks in process control systems is to learn unsupervised clustering models containing the pair-wise relationship between variables of a process. *Attack Detection:* Using these models it is possible to identify anomalies that do not fit the clusters

[47, 52]. *Attak Location:* They consider sensor attacks. *Validation:* The authors used the Tennessee Eastman process simulation to validate the proposed approach for several types of attacks.

These approaches are non-parametric, which have the advantage of creating models of the physical process without a priori knowledge of the physics of the process; however, a non-parametric approach does not have the fidelity to the real physics of the system as an LDS or AR model will have, in particular when modeling the time-evolution of the system or the evolution outside of a steady state.

Detecting Safety Violations and Response. Another paper that proposes control reconfiguration is McLaughlin [67]. This paper tackles the problem of how to verify that control signals u_k will not drive the system to an unsafe state, and if they do, to modify the control signal and produce a reconfiguration control that will prevent the system from reaching an unsafe state. As such, this is one of the few papers that considers a reconfiguration option when an attack (or possible safety violation) is detected. The proposed approach, C^2, mediates all control signals u_k sent by operators and embedded controllers to the physical system. *Attack Detection:* C^2 considers multiple systems with discrete states and formal specifications, as such this approach is better suited for systems where safety is specified as logical control actions instead of systems with continuous states (where we would need to use system identification to learn their dynamics). This approach is most similar to the attack on control signals in Figure 2. However, their focus is not to detect if $u_k \neq \mathcal{K}(y_k)$, but to check if u_k will violate a safety condition of the control signal or not. As such, their approach is most similar to using the *Safety Check* block we introduced in Figure 6. *Attack Location:* McLaughlin mentions that "the approach can prevent any unsafe device behavior caused by a false data injection attack, but it cannot detect forged sensor data," and later in the paper we find "C^2 mitigates all control channel attacks against devices, and only requires trust in process engineers and physical sensors." This is a contradiction, and the correct statement to satisfy the security of their model is the latter. As such C^2 assumes trusted sensors and trusted actuation devices (specifically stating trusted actuators is a missing trust assumption in their model). C^2 is related to traditional safety systems for control like safety interlocks, and not necessarily malicious attacks as there does not seem to be a difference between preventing an unsafe accidental action to an unsafe malicious action. *Validation:* There are three main properties that C^2 attempts to hold: (1) *safety* (the approach must not introduce new unsafe behaviors, i.e., when operations are denied the "automated" control over the plant should not lead the plant to an unsafe state), (2) *security* (mediation guarantees should hold under all attacks allowed by the threat model), and (3) *performance* (control systems must meet real time deadlines while imposing minimal overhead). C^2 was tested using simulations and real PLCs for six different case studies.

Detecting Malicious Control Commands. There is other related work in trying to understand consequences of potentially malicious control commands from the control center. Their goal is to understand safe sequences of commands and commands that might create problems to the system. *Attack Location:* These attacks correspond (logically) to the attack on control signals in Figure 2 [51, 58, 82]. *Attack Detection:* Lin et al. [58] use contingency analysis to predict the consequences of control commands, and Mitra et al. [73] combine the dynamics of the system with discrete transitions (finite state machines) such as interruptions. Using set theory, they show it is possible to determine the set of safe states, the set of reachable states, and invariant sets; therefore, if there is not an input that can drive the states out of the safety set, the model is safe. Finding these sets requires some relaxations and a good knowledge of the behavior and limitations of the system. Finally, attacks are detected using a stateless IDS, that checks the validity of the control commands. *Validation:* The approach is validated by using simulations.

Active Monitoring for Sensors. Active monitoring has also been used to verify the freshness and authenticity of a variety of sensors [93] and video cameras [106]. *Attack Detection:* PyCRA [93] uses an LDS model to predict the response of sensors and to compute the residual r_k, which is then passed to a stateful χ^2 anomaly detection statistic. The attacker in PyCRA has a physical actuator to respond to the active challenge. *Validation:* The evaluation of the proposal focuses on computing the trade-off between false alarms and probability of detection (i.e., ROC curves).

Another active monitoring approach suggests *visual challenges* [106, 107] to detect attacks against video cameras. *Anomaly Detection:* In particular, a trusted verifier sends a visual challenge such as a passphrase, Quick Response (QR) code, or random tweets to a display that is part of the visual field of the camera, and if the visual challenge is detected in the video relayed by the camera, the footage is deemed trustworthy. The paper considers an adversary model that knows all the details of the system and tries to forge video footage after capturing the visual challenge. The authors use the CUSUM statistic to keep track of decoding errors. *Validation:* Using real data from a test-bed, the authors evaluated several visual challenges and investigated their advantages.

Electricity Theft. There is also work on the problem of electricity theft detection by monitoring real traces from electricity consumption from deployed smart meters [66]. *Attack detection:* To model the electricity consumption the authors use ARMA models, which are output-only models similar to those in Equation (1). Since their detection is not done online (similar to the video forensics case), the detection test is not stateless but stateful (an average of the residuals), where the detector can collect a lot of data and is not in a rush to make a quick decision. *Attack location:* The attacker has compromised one sensor (the smart meter at their home) and sends false electricity consumption. *Validation:* The evaluation metric is the largest amount of electricity that the attacker can steal without being detected and the trade-off between the false positive rate and the cost of an attack. Authors used a real-data set to validate their results.

Platooning. Sajjad et al. [88] consider the control of cars in automated platoons. *Attack Detection:* They use LDS to model the physical system and then use a *stateful* test with a fixed window of time to process the residuals. *Attack Location:* Authors consider attacks in sensors and actuators. *Validation:* To evaluate their system they show via simulation that when attacks are detected, the cars in the platoon can take evasive maneuvers to avoid collisions.

4.4 Miscellaneous Domains

There are several other papers leveraging physics-based attack detection that have been published in conferences or journals not traditionally associated (or focused) on control theory, power systems, or cyber security. We now summarize some of these results.

Video Forensics. Conotter et al. [19] propose to look at the geometry and physics of free-falling projectiles to check if the motions of a moving object in videos are realistic or fake. *Attack detection:* The proposed algorithm to detect implausible trajectories of objects follows: First, describe a simplified 3D physical model of the expected trajectory and a simplified 2D imaging model. Then, determine if the image of the trajectory of a projectile motion is consistent with the physical model. A contribution of the paper is to show how a 3D model can be directly created from the 2D video footage. Once a 3D model is created, it can be used to check against the physical model to detect any deviations. The attacker is someone who uses sophisticated video editing tools to manipulate a video of, for example, a person throwing a basketball to create a perfect, spectacular shot. In this case, the forger has access to the 2D video footage and can manipulate, re-process it. The paper does not focus on how the forgery is done but assumes that a video can be either fake or real, and the goal of the proposed approach is to determine the authenticity of each video. However, note that only naive attackers were considered here. If the forger is aware of such detection

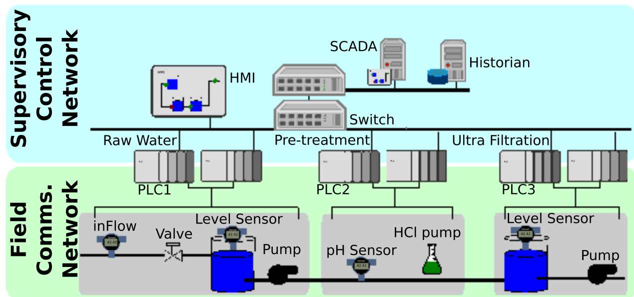

Fig. 10. Communication between actuators or sensors to PLCs is achieved by field communication protocols. Control between PLCs or between PLC and a central control server is achieved with supervisory industrial protocols. This network is part of a testbed we use for our experiments.

mechanism, then it will try to manipulate the 2D image to conform to the real 3D model. *Validation:* The evaluation metric computes the mean error between the pair of representations of the projectile motion using Euclidean distance; so it is a stateful test. The reason for using this test (and not change detection statistics) stems from the fact that forgery detection does not need to be done in real-time, but it is mostly done after the fact.

Medical Devices. Detection of attacks to medical devices is also a growing interest [35, 36]. *Attack location:* Hei et al. [35] study overdose attacks/accidents for insulin pumps. *Attack Detection*: They employ a supervised learning approach to learn normal patient infusion patterns with dosage amount, rate, and time of infusion. The model of their physical system is done through a Support Vector Regression (SVR). Again, similar to all the papers reviewed in this miscellaneous section focusing on off-line anomaly detection, the detection test is an average of the residuals. More specifically, they use the Mean Squared Error measuring the difference between the predicted and the real value before raising an alert. *Validation:* They validated the impact of the attacks and the effectiveness of their detection mechanism using real applications.

Critical State Analysis. Carcano et al. [12] propose a safety monitoring system similar to C^2 but without mediating control commands (and using the control command u_k to predict the next state $\hat{y}_k$ to see if it violates a safety condition) or proposing any reconfiguration when a safety issue is detected. The proposed concept is to monitor the state of a system and raise alerts whenever it is in a critical state (or approaching a critical state). *Attack Detection:* the approach measures the distance of sensor measurements y_k to a critical state y^c: $d(y_k, y^c)$. They do not learn the dynamics of the physical system and this can have serious consequences as, for example, the power grid can change the distance to a critical state almost immediately, whereas chemical processes such as growing bacteria in anaerobic reactors can take days to drive a system state to an unsafe region. An alert is risen whenever the system is in a critical state and also log the packets that led the system to that state for forensic purposes. They only monitor y_k not u_k, which as we will show, is a suboptimal approach. *Attack Location:* Because the authors monitor Modbus commands, it is likely that their sniffer is installed at the Supervisory Control Network of Figure 10, and as we will show, this assumes a trusted PLC. They also assume trusted sensors. The simulated attacks consist of legitimate control commands that drive the system to unsafe states; as such, these attacks are easy to detect. *Validation:* they monitor the number of false alarms and the true positive rate. The detection algorithm can have missed positives (when an attack happened and was not detected) because of packet drops, but it is not clear what a false alarm is in their case (it appears to be a critical state caused by legitimate control actions).

Table 2. Summary of Taxonomy of Related Work on Physics-based Attack Detection in Control Systems

		[80] Murguia *et al.*	[73] Mishra *et al.*	[102] Teixeira *et al.*	[8] Bai, Gupta	[78] Mo *et al.*	[76] Mo, Sinopoli	[9] Bai *et al.*	[72] Miao *et al.*	[39] Hou *et al.*	[25] Eyisi *et al.*	[77] Mo *et al.*	[85] Pasqualetti *et al.*	[105] Teixeira *et al.*	[55] Kwon *et al.*	[104] Teixeira *et al.*	[24] Do *et al.*	[4, 5] Amin *et al.*	[96] Smith	[47] Kerns *et al.*	[11] Bobba *et al.*	[92] Sandberg *et al.*	[58] Liang *et al.*	[30] Giani *et al.*	[22] Dan, Sandberg	[52] Kosut *et al.*	[48] Kim, Poor	[23] Davis *et al.*	[99] Sridhar, Govindarasu	[53] Koutsandria *et al.*	[68] Mashima *et al.*	[62, 63] Liu *et al.*	[84] Parvania *et al.*	[60] Lin *et al.*	[95] Shoukry *et al.*	[32] Hadziosmanovic *et al.*	[14] Cardenas *et al.*	[113] Wang *et al.*	[69] McLaughlin	[91] Sajjad *et al.*	[54] Krotofil *et al.*	[108] Valente, Cardenas	[110] Vukovic, Dan	[79] Morrow *et al.*	[21] Cui *et al.*	[13] Carcano *et al.*	[36] Hei *et al.*	[49] Kiss *et al.*	[20] Conotter *et al.*
	Venue	Control Theory																			Power Systems										Cyber-Security												Misc.						
Attack Detection	**Detect. Stat.**																																																
	stateless	●	-	●	-	-	-	●	●	●	◐	●	●	●	●	●	-	●	◐	●	●	●	-	●	●	●	●	●	-	●	-	●	◐	●	-	●	-	◐	◐	-	●	-	●	●	●	●	-	●	-
	stateful	●	⊛	-	◐	⊛	⊛	-	-	-	-	-	-	-	-	-	●	-	-	-	-	-	⊛	-	-	-	-	-	⊛	-	●	-	-	-	⊛	-	●	-	-	●	●	●	-	-	●	-	⊛	-	●
	Prediction																																																
	AR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	●	-	-	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-
	SLS	-	-	◐	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	●	●	●	●	●	●	●	◐	-	-	-	●	-	-	-	-	-	-	-	-	-	-	◐	◐	●	-	-	-	-
	LDS	●	●	-	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	●	-	●	-	-	●	-	-	-	-	-	-	-	-	-
	other	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	●	●	-	-	◐	◐	-	-	-	●	●	-	●	●	-	-	-	●	◐	●	●
Attack Location	sensors	●	●	●	●	●	●	-	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	-	●	●	-	-	●	●	●	◐	-	-	●	●	-	●	●	◐	-	●	●
	actuators	-	-	-	-	●	●	●	-	-	-	●	●	●	●	●	●	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	●	-	-	◐	-	-	-	-	-	-	-	-	-	-	●	-	-
	controllers	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	●	-	-	-	-	●	-	-	-	-	●	●	-	-	●	-	-	◐	-	-	-
Validation	**Metrics***																																																
	impact	●	●	-	●	-	-	●	-	●	●	-	●	●	●	●	-	-	-	●	-	●	●	-	●	-	●	-	-	●	●	●	-	●	-	-	-	-	-	●	-	-	●	-	●	-	●	-	●
	statistic	●	-	●	-	●	-	-	●	●	-	-	●	●	●	●	-	●	-	-	-	●	●	-	-	●	-	-	-	●	-	●	-	-	-	-	-	-	-	-	●	●	●	-	●	-	-	●	●
	TPR	-	-	-	-	●	●	-	-	-	-	●	-	-	-	-	●	-	-	-	-	-	-	●	-	●	-	-	●	-	-	-	-	-	●	●	-	●	-	-	-	-	-	-	-	●	-	-	-
	FPR	-	-	-	-	●	-	-	-	-	-	●	-	-	-	-	●	-	-	-	-	-	-	-	-	●	-	-	●	-	●	-	-	-	●	-	-	●	-	-	-	-	-	-	-	●	-	-	-
	Time-Detect	-	●	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	●	-	-	-	-	●	-	-	●	-	-	-	-
	Implement.																																																
	simulation	●	●	●	●	●	●	●	●	●	●	●	●	●	●	-	●	-	●	●	-	●	●	-	●	●	●	●	●	◐	-	●	-	●	-	-	●	●	-	●	●	-	●	●	●	-	-	●	-
	real data	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	●	-	-	-	-	-	-	-	-	-	◐	●	-	●	-	-	-	-	●	-	-	-	-	-	●	-	-	-	-	●	-	●
	testbed	-	-	-	-	-	-	-	-	-	-	-	-	-	-	●	-	-	-	●	-	-	-	-	-	-	-	-	-	-	-	-	◐	-	●	●	-	-	●	-	-	-	-	-	-	●	-	-	-
	Monitoring	◇	◇	◇	◇	◇	◇	◇	◇	◇	◇	◆	◇	◇	◇	◇	◇	◇	◇	◇	◇	◇	◇	◇	◇	◇	◇	◆	◇	◇	◇	◇	◇	◇	◆	◇	◇	◇	◇	◇	◇	◆	◇	◆	◇	◇	◇	◇	◇

Legend: ●: feature considered by authors, ◐: feature not explicitly stated or exhibits ambiguity, ⊛: a windowed stateful detection method is used, ◇: passive monitoring, ◆: active monitoring, †: attacks are made on the communication layer, ‡: also considers physical attacks, *Evaluation options have been abbreviated in the table: Attack Impact, Statistic Visualization, True Positive Rate, False Positive Rate.

5 DISCUSSION

We apply our taxonomy to previous work in Table 2. The rows of the table correspond to our taxonomy, and the columns of the table correspond to *main emphasis* of the publication venue. We make the following observations: (1) the vast majority of prior work uses stateless tests; (2) most control and power grid venues use LDS (or their static counterpart SLS) to model the physical system, while computer security venues tend to use a variety of models; several of them are non-standard and difficult to replicate by other researchers; (3) there is no consistent metric used to evaluate proposed attack-detection algorithms; (4) most papers focus on describing attacks to specific devices (i.e., devices that are not trusted) but they do not provide a fine-grain trust model that can be used to described what can be detected and what cannot be detected when the adversary is in control of different devices; and (5) no previous work has validated their work with all three options: simulations, testbeds, and real-world data.

5.1 General Shortcomings

(1) **Lack of Trust Models.** Most papers do not describe their trust models with enough precision. Information exchanged between field devices (sensor to controller and controller to actuator in Figure 1) is communicated through a different channel from information that is exchanged between controllers or between controller and the supervisory control center. Papers that monitor network packets in the supervisory control network [31]

implicitly assume that the controller (PLC) they monitor is trusted, otherwise the PLC could fabricate false packets that the monitor expects to see, while at the same time sending malicious data to actuators (what Stuxnet did). Thus, we need to monitor the communication between field devices to identify compromised PLCs in addition to monitoring supervisory control channels to identify compromised sensors or actuators.

(2) **No Consistent Evaluation.** There is no common evaluation metric used across multiple papers. Some papers [12, 111] measure the accuracy of their anomaly detector by looking at the trade-off between the false-alarm rate and the true positive rate (metrics that are commonly used in machine-learning, fault-detection, and some intrusion detection papers), while others [31] argue that measuring the true positive rate is misleading in a field that has not enough attack samples, so they focus only on measuring the fidelity of their models (i.e., minimizing the false alarms). In addition, most papers focusing on false data injection for state estimation in the power grid and most papers in control theory tend to focus on developing new undetected attacks, and ignore completely the number of false alarms. Another concern is that several papers do not consider the delay in detecting an attack, and as we can see in our table, we need more research focusing on improving the time-to-detect metrics, so that attacks cannot cause irreparable damage by the time they are detected. It is important to note that most of the papers that consider stateful detection mechanisms like the CUSUM are actually using statistical tools designed precisely to minimize the time to detect an attack. As such they are implicitly including the time-to-detect as a metric.

(3) **No Comparison among Different Models and Different Tests.** There is no systematic publication record that builds upon previous work. While previous work has used different statistical tests (stateless vs. stateful) and models of the physical system to predict its expected behavior (AR vs. LDS), so far they have not been compared against each other, or if a given combination of physical models with the appropriate anomaly detection test is the best fit.

(4) **Experiments.** We have not seen a detailed discussion on the different considerations, advantages, and disadvantages of using real data from operational systems, testbeds, or simulations. Each of these experimental scenarios are different and provide unique insights as well as unique limitations for physics-based detection algorithms.

Suggested Improvements. To address the first limitation, we propose a set of guiding principles for discussing trust models for attack detection in control systems in Section 6.1. To address the last two points, we introduced a new evaluation metric [105], which we describe in Section 6.2.

6 IMPROVING PHYSICS-BASED ATTACK DETECTION

6.1 Trust Assumptions

Understanding the general architecture between actuators, sensors, controllers, and control centers is of fundamental importance to analyze the implementation of a monitoring device and most importantly, the trust assumptions about each of these devices, as any of these devices (actuators, sensors, PLCs, or even the control center) can be compromised.

Control systems have in general a layered hierarchy [114], with the highest levels consisting of the **Supervisory Control Network (SCN)** and the lowest levels focusing on the **Field Communications Network (FCN)** with the physical system, as shown in Figure 10. A survey of communications in industrial control systems can be found in Gaj et al. [26].

If we were to deploy our anomaly detection system in the SCN (which typically has network switches with mirror ports making it the easy choice), then a compromised PLC can send

Table 3. Detectability of Attack Depending on Trust in Components

Component Trust			Detection	
PLC	Sensor	Actuator	possible	Comment
✓	-	-	-	Bad actuation and bad sensing
-	-	✓	-	False sensing justifies bad controls
-	✓	-	~	Attack effects observable
✓	-	✓	✓	Attack effects observable
✓	✓	-	✓	Attack effects observable
-	✓	✓	✓	Bad command detection
✓	✓	✓	✓	No attack possible

✓ = trusted/detection possible, - = untrusted/detection not possible, ~= cannot detect zero-dynamics attacks.

manipulated data to the FCN, while pretending to report that everything is normal back to the SCN. In the Stuxnet attack, the attacker compromised a PLC (Siemens 315) and sent a manipulated control signal u^a (which was different from the original u, i.e., $u^a \neq u$) to a field device. Upon reception of u^a, the frequency converters periodically increased and decreased the rotor speeds well above and below their intended operation levels. While the status of the frequency converters y was then relayed back to the PLC in the field communications layer, the compromised PLC reported a false value $y_a \neq y$ to the control center (through the SCN) claiming that devices were operating normally.

By deploying our network monitor at the SCN, we are not able to detect compromised PLCs (unless we are able to correlate information from other trusted PLCs), or unless we receive (trusted) sensor data directly.

A number of papers we analyzed did not mention where the monitoring devices will be placed, which makes it difficult to analyze the author's trust model. For example, analyzing the DNP3 communications standard [57, 58] implicitly assumes that the monitoring device is placed in the SCN, where DNP3 is most commonly used, and this security monitor will thus miss attacks that send some values to the SCN, and others to the FCN (such as Stuxnet). Therefore, such papers implicitly assume that the PLC is reporting truthfully the measurements it receives, and the control commands it sends to actuators. This weak attacker model limits the usefulness of the intrusion detection tool.

To mitigate such restrictions, we argue that anomaly detection monitors should (also) be used at the FCN to detect compromised PLCs, actuators, and sensors. Assuming the monitor is placed in the FCN, the selection of trusted components determines the kind of attacks that can be detected (see Table 3). Our analysis shows that as long as you trust two components in the loop, it is possible to detect an attack on the remaining component. If we trust the sensors but do not trust either the actuators or the PLCs, then we can still detect attacks, unless they are zero-dynamic attacks [83, 102, 103] (although not all physical systems are vulnerable to these attacks). Finally, if we only trust the actuator (or only the PLC), the attacks could be completely undetected. We note that while there are still some attacks that cannot be detected, we can still detect more attacks than at the SCN.

To illustrate some advantages of monitoring the FCN, we show experimental results obtained from a water treatment testbed.

6.1.1 Minimizing Trust Assumptions by Developing a Security Monitor in the Field Layer of Industrial Control Systems. The *Secure Water Treatment (SWaT)* testbed we use for our experiments is a water treatment plant consisting of six main stages to purify raw water. The testbed has a total

(a) Reverse Osmosis stage. (b) Raw Water stage.

Fig. 11. Illustrations of the SWaT testbed.

of 12 PLCs (six main PLCs and six in backup configuration to take over if the main PLC fails). The general description of each stage is as follows: *Raw water storage* is the part of the process where raw water is stored and it supplies water to the water treatment system. It consists of one tank, an on/off valve that controls the inlet water, and a pump that transfers the water to the ultra filtration (UF) tank. In *Pre-treatment* the Conductivity, pH, and Oxidation-Reduction Potential (ORP) are measured to determine the activation of chemical dosing to maintain the quality of the water within some desirable limits. *Ultra Filtration* is used to remove the bulk of the feed water solids and colloidal material by using fine filtration membranes that only allow the flow of small molecules. Then, the remaining chlorines are destroyed in the *Dechlorinization* stage, using ultraviolet chlorine destruction unit and by dosing a solution of sodium bisulphite. The *Reverse Osmosis* (RO) system is designed to reduce inorganic impurities by pumping the filtrated and dechlorinated water with a high pressure (see Figure 11(a)). Finally, the *RO final product* stage stores the RO product (clean water). Each stage is controlled by two PLCs (primary and backup); the primary and backup PLC for the raw water stage can be seen in Figure 11(b). The PLC receives the sensor information (water level and water flow for stage 1) and computes the corresponding control actions. The field devices, i.e., actuators/sensors, send and receive 4–20mA signals that must be converted back and forth to their corresponding physical value.

The network of the testbed (illustrated in Figure 10) uses the Common Industrial Protocol (CIP) [11] as the main data payload for device communications at the SCN, while a device-and-vendor dependent I/O implicit message is used at the FCN. The payloads are encapsulated following the Common Packet Format of the EtherNet/IP specification [80] and transported through any of the two available physical layers: either wired over IEEE 802.3 Ethernet or wireless using IEEE 802.11.

The availability of a semantically rich network protocol like CIP at the SCN layer facilitates deep-packet inspection, because parsing and extracting semantically meaningful values is fairly straightforward; however, performing deep-packet inspection at the Field layer means working with low-level data where values are exchanged without standard units of measurement, and where the protocol is not publicly available. This difference is one of the biggest challenges in deploying security monitors in the field layer and one we tackle next.

I/O implicit messages are device and vendor dependent (Allen-Bradley in this deployment), and because the specification is not publicly available, we used Wireshark [116] together with the Testbed's Control Panel and Electrical Drawings manual to develop the exact structure of the EtherNet/IP-encapsulated I/O implicit messages.

We identify three different vendor and device-dependent I/O implicit messages corresponding to each of the three types of signals the field devices send and receive (see Table 4): analog input,

Table 4. I/O Implicit Messages

I/O Message	Signal size (bits)	No. of signals	Avg. Freq. (ms)
Digital Input	1	32	50
Digital Output	1	16	60
Analog Input	16	12	80

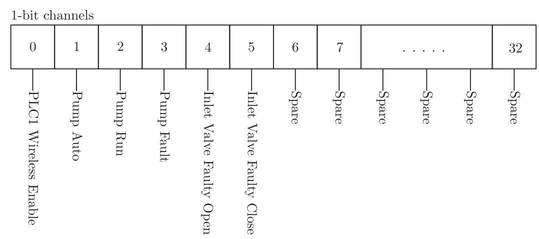

Fig. 12. Digital input module with 32 input signals (1-bit signals) for the raw water storage stage.

digital input, and digital output signals. Figure 12 shows the I/O implicit message for the digital input signals. It is a stream of 32 bits, corresponding to each of the digital inputs signals. The *spare* channels are those not in use by the current deployment. The digital outputs are grouped in a 16-bit stream (1 bit per signal), while the analog inputs are grouped in a 24-byte stream with 16 bits per signal.

The I/O implicit messages representing the analog signals are sent by the field devices to the PLC with an average frequency of 80ms. They transport the numeric representation of the 4–20mA signals measured by the analog sensors. To scale back and forth between the 4–20mA signal to the real measurement, we use the Equation (3). The constant values depend on the deployment and the physical property being measured:

$$Out = (In - RawMin) * \frac{EUMax - EUMin}{RawMax - RawMin} + EUMin. \tag{3}$$

We developed a command-line interpreter (CLI) application that includes a library of attacks and a network monitoring module implementing stateful and stateless detection mechanisms. The attack modules are capable of launching diverse spoofing and bad-data-injection attacks against the sensor and actuator signals of the testbed. The attack modules can be loaded, configured, and run independently of each other, allowing to attack sensors/actuators separately. The attack modules also can be orchestrated in teams to force more complex behaviors over the physical process, while maintaining a normal operational profile on the Human Machine Interface (HMI). The CLI application consists of 632 lines of Python [118] 2.7 code and its only external dependencies are Scapy and NetFilterQueue.

Making use of Scapy [119], we developed a new protocol parser for the Allen-Bradley proprietary I/O implicit messages used for signal communication between the field devices and the PLCs, and for the EtherNet/IP Common Packet Format wrapper that encapsulates it. Scapy was also used to sniff, in real-time, the sensor readings and actuation commands from the EtherNet/IP-encapsulated messages and to inject them with fake data [104]. Our software calculates the data integrity checksums used by the Transport Layer protocol in use; the FCN makes use of User Datagram Protocol (UDP) for the transport of EtherNet/IP I/O implicit messages among field devices.

To avoid duplication of packets and/or race conditions between original and injected packets, we employed the NetFilterQueue [117] Python bindings for libnetfilter queue to redirect all the

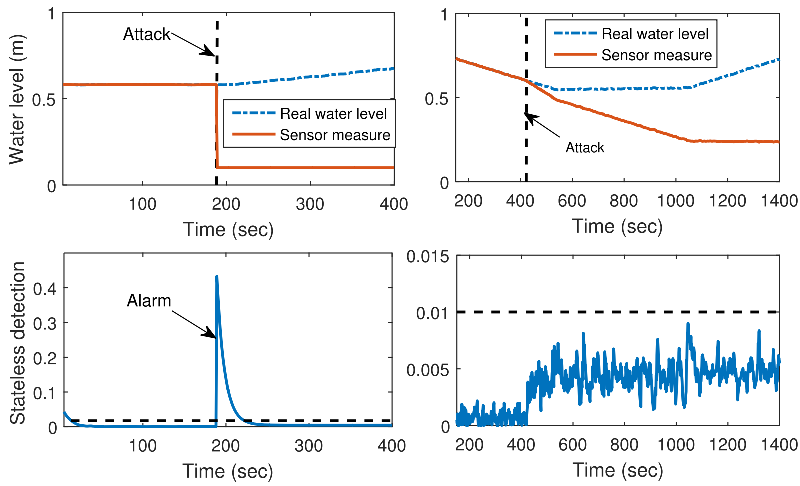

Fig. 13. (Left) A sensor attack (in orange) starts at time 200s. A false sensor value of 0.1m forces the PLC to turn on the pump to fill the tank with water. The real height of water in the tank starts increasing (blue) and will continue until it overflows the tank. (right) A more intelligent attack that remains undetected by changing the sensor measurement slowly. Its impact is not critical due to the control actions.

EtherNet/IP I/O messages between PLC and the field devices to a handling queue defined on the PREROUTING table of the Linux firewall *iptables*. The queued packets can be modified using Scapy and the previously mentioned message parser and finally released to reach their original destination, e.g., PLC or field devices. Likewise, this technique allowed us to avoid disruptions on the sequence of EtherNet/IP counters and injection of undesirable perturbations in the EtherNet/IP connections established between field devices.

We now illustrate how our tool can be used to launch and detect attacks in the testbed.

Attacking the Water Level. The goal of the attacker is to deviate the water level in a tank as much as possible until the tank overflows.

To detect these spoofed sensor values, we use an LDS model of the water level. In particular, we use a mass balance equation that relates the change in the water level h with respect to the inlet Q^{in} and outlet Q^{out} volume of water, given by $Area\frac{dh}{dt} = Q^{in} - Q^{out}$, where $Area$ is the cross-sectional area of the base of the tank. Note that in this process the control actions for the valve and pump are On/Off. Hence, Q^{in} or Q^{out} remain constant if they are open and zero otherwise. Using a discretization of 1s, we obtain an estimated model of the form

$$h_{k+1} = h_k + \frac{Q_k^{in} - Q_k^{out}}{Area}.$$

Note that while this equation might look like an AR model, it is in fact an LDS model, because the input $Q_k^{in} - Q_k^{out}$ changes over time, depending on the control actions of the PLC (open/close inlet or start/stop pump). In particular, it is an LDS model with $x_k = h_k$, $u_k = [Q_k^{in}, Q_k^{out}]^T$, $B = [\frac{1}{Area}, -\frac{1}{Area}]$, $A = 1$, and $C = 1$.

We start by using a stateless anomaly detection mechanism to identify attacks. Figure 13 (left) shows a sensor attack (in orange) starting at time 200s. While the real height of the water in the tank is 0.5m, a false sensor value of 0.1m forces the PLC to turn on the pump to fill the tank with water. The real height of water in the tank starts increasing (blue) and will continue until it

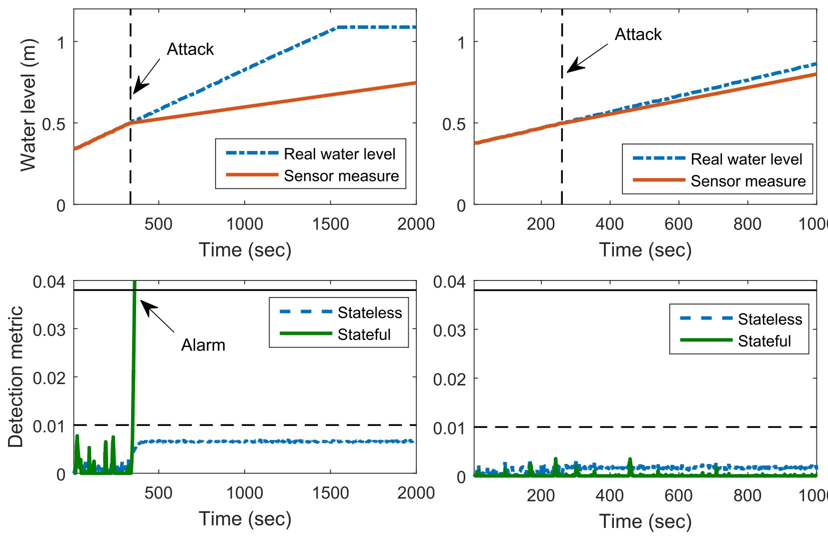

Fig. 14. (Left) Undetected attack that seeks to overflow the tank. Note that using stateless detection it is not possible to detect and the water is spilled. Stateful detection accumulates the residuals fast enough to detect the attack. (Right) The attack is designed to make it stealthy for both detection mechanisms. However, the impact (deviation from the HIGH value) is very small.

overflows the tank. This abrupt change observed by our attack-detection tool, from 0.5m to 0.1m in the height of the tank in an instant does not match the physical equations of the system, and therefore the residual value (lower left plot) will increase way above the dotted line that represents the threshold to raise an alert.

As we can see, it is very easy to create attacks that can be detected, and this poses a challenge for designing good evaluation metrics and good attacks. If we use the detection rate (true positive rate) as a metric for these attacks, then we would always get 100% detection rate.

However, for any physical system a sophisticated attacker can spoof deviations that follow relatively close to the "physics" of the system while still driving the system to a different state. Figure 13 (right) shows an attack starting at time 400s that slowly starts to change the false sensor value (orange) forcing the real height of the water in the tank to grow; however, the anomaly detection statistic (bottom right) does not reach the threshold necessary to raise an alarm.

We can also compare the performance of a CUSUM stateful vs. a stateless test for these types of undetected attacks. Figure 14 (left) shows how an attack that tries to fake a sensor signal growing slower from its real value can bypass a stateless anomaly detection statistic and overflow the tank; however, it will be detected by the CUSUM statistic. Figure 14 (right) shows that if the attacker wants to avoid being detected by the CUSUM statistic, then the amount of deviation it can inject to the system is so small, that it cannot force an overflow of the tank (i.e., it cannot drive the real water height to 1.1m). In short, the selection of the appropriate anomaly detection statistic can limit the ability of an attacker to damage the system, but we need a systematic way to quantify the effectiveness of these defenses.

6.2 Towards Better Evaluation Metrics

One of the differences between detecting attacks in control systems when compared to detecting attacks in general IT systems is that researchers do not have readily available data from attacks in the wild. Even if we test our algorithms on the few known examples (like Stuxnet), they are domain specific, and it is not clear they will give insights into the evaluation other than to show that we can detect Stuxnet (which can be easily detected *ex post*). For that reason, researchers need to generate novel attacks in their papers, and the question we would like to address in this section is how to create attacks that are general enough to be applicable across multiple industrial control domains but that will also allow us to define an evaluation metric that is fair (and that is not biased to detect the specific attacks from the researchers).

To motivate the need of a new metric, we now discuss the challenges and limitations of previously used metrics.

Measuring the True Positive Rate is Misleading. To obtain the true positive rate of a detection algorithm, we need to generate an attack that will be detected. It is not clear if there can be a principled way of justifying the generation of an attack that will be detected as this implies our attacker is not adaptive and will not attempt to evade our detection algorithms. Publications using the true positive rate [12, 111] generate their attacks as random signals (e.g., a sensor reporting random values instead of reporting the true state of the physical system). This type of non-strategic random failure is precisely what the fault-detection community has been working on for over 40 years [115]; with those attacks, we are not advancing the state of the art on attack-detection, but rather reinforcing the fact that fault-detection works when sensor or control signals fail in a non-malicious way.

Model Fidelity is an Incomplete Metric. One of the first papers to articulate why measuring in a meaningful way the true positive rate for control systems is hard is the work of Hadziosmanovic et al. [31]. Having summarized the reasons why measuring the true positive rate can be misleading, they focus instead on understanding how accurately their AR system models the real-world system and identifying the cases where it fails. They are more interested in understanding the model fidelity than in specific true-/false-alarm rates. However, understanding the model fidelity is implicitly looking at the potential of false alarms, because deviations between predictions and observations during normal operations are indicators of false alarms. While this is a good approach for the exploratory data analysis done in the paper, it might be misunderstood or applied incorrectly by future researchers. The anomaly detection rule of "*never raise an alert*" will have zero false alarms—i.e., perfect fidelity—but it never detects any attack.

Ignoring False Alarms Does not Provide a Complete Evaluation. As we discussed before, the line of research started by false data injection attacks for state estimation in the power grid [60, 61] focuses on developing new ways to find attacks or to find new undetectable attacks; however, they tend to ignore the evaluation of the system under normal conditions (the false-alarm rate). A similar emphasis on attack detection and on identifying undetectable attacks but ignoring false alarms can be seen in the control theory community [83]; at the end of the day, you can detect all attacks by generating an alert at every single time-step k, but this will give raise to an unmanageable number of false alarms.

Lessons From The Last Three Attacks in Section 6.1. If we had evaluated our anomaly detection algorithm against using a traditional intrusion detection metric like ROC curves, and our attack examples consisted of the last three attacks presented in the previous section (a stealthy attacker), then we would have had a 0% detection rate; that is, our ROC curve would be a flat line along the x-axis with a 0% value in the y-axis (Figure 17 (left)). This problem is not unique to ROC curves, most popular metrics for evaluating the classification accuracy of intrusion detection systems can be shown to be a multi-criteria optimization problem between the false-alarm rate, and the true positive rate [14], and all of them depend on the ability of a system to detect some attacks.

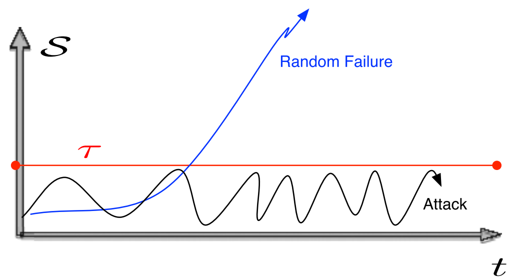

Fig. 15. Difference between a fault and an attack: a sophisticated attacker will remain undetected by maintaining the anomaly detection statistic $\mathcal{S}$ below the threshold τ to avoid raising alarms.

To obtain the true positive rate of a detection algorithm, we need to generate an attack that will be detected, and it is not clear if there is a principled way of justifying that to evaluate a system we need to generate attacks that will be detected, as this implies that the adversary is not adaptive and will not attempt to evade our detection algorithms.

In the previous section, we showed that for any anomaly threshold τ, a "smart" attacker can always launch an attack that keeps the anomaly detection statistic below this threshold, and therefore this "smart" attacker can always launch attacks that will not be detected (i.e., the attacker can create a variety of attacks that will have a 0% detection rate). Figure 15 illustrates this problem. In this figure, an anomaly detection statistic $\mathcal{S}$ keeps score of the "anomalous" state in the system: if $\mathcal{S}$ increases beyond the threshold τ, it will raise an alarm. Random failures are expected to increase the anomaly score, but a sophisticated attacker that knows about this anomaly detection test will be able to remain undetected.

The question that we need to answer here is then, **How much can the attacker affect the system while remaining undetected?**

In addition to a metric that quantifies how much the attacker can affect the system without being detected, we need to consider a metric that shows the trade-offs involved. Most of the work in control theory and power system conferences ignore false-alarm rates in their analyses [60, 61, 83]; however, at the end of the day, you can detect all attacks by generating an alert at every single time-step k, but this will give rise to an unmanageable number of false alarms, so we need to illustrate the inherent trade-off between security and false alarms (usability).

In conclusion, the traditional trade-off between false alarms and detection rate is not a good fit for our problem; however, focusing solely on model fidelity will not give us any indication of what an attacker can do. Ignoring false alarms prevents assessment of the practicality and usability of the system.

Design Options for Metrics. Looking again at our literature review, the majority of previous work uses a model of the physical system (*LDS* or *AR*) to generate an expected value $\hat{y}_k$. This prediction is then compared to the sensor measurements y_k to generate a residual $r_k = |\hat{y}_k - y_k|$. We test if $r_k > \tau$, where τ is a threshold we can adjust to lower false alarms while still hoping to achieve good detection.

A *stateless* test generates an alarm if $r_k > \tau$, where τ is a threshold we can adjust to lower false alarms while still hoping to achieve good detection. A *stateful* test instead will compute an additional statistic S_k that keeps track of the historical changes of r_k and will generate an alert if $S_k \geq \tau$ (another appropriately chosen threshold).

We can clearly see that increasing the threshold will reduce the number of false alarms; however, what do we give up by reducing the number of false alarms? Traditionally the trade-off for reducing

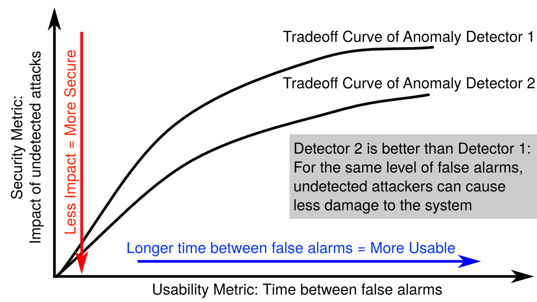

Fig. 16. Illustration of our proposed tradeoff metric in Reference [105]. The y-axis is a measure of the worst the attacker can do while remaining undetected, and the x-axis represents the expected time between false alarms $E[T_{fa}]$. Anomaly detection algorithms are then evaluated for different points in this space.

the number of false alarms is a reduced true positive rate, but as we discussed before, this is not a good metric for our case. Notice that, if the threshold is too low, an attacker has to produce attacks where y_k will be similar from the expected behavior of our models, but if it is too high, the attacker has more leeway to deviate y_k and damage to the system without raising alarms. We argue that the metric that we need is one that shows the trade-off between the number of false alarms and the ability to minimize the negative consequences of undetected attacks.

Summary. A *classification accuracy* metric of an anomaly detection algorithm $\mathcal{A}$ needs to capture two things: (1) the ability of $\mathcal{A}$ to detect attacks (we call this a *security metric*) and (2) the ability of $\mathcal{A}$ to label correctly *normal* events so that it does not raise too many false alarms (we call this a *usability metric*). The *security metric* and the *usability metric* represent a trade-off that needs to be balanced (lower false-alarm rates typically means lower ability to detect attacks), and therefore we need to include both (the security metric and the usability metric) in a trade-off plot.

6.3 New Evaluation Metric

It is clear that we need to find a consistent way to evaluate and compare different anomaly detection proposals, but so far there is little research trying to address this gap. We have recently proposed a new evaluation metric that takes into account the usability and security factors for physics-based attack detection algorithms [105]. We have analyzed the trade-off between the impact of the worst attack the adversary can launch while remaining undetected (y-axis) and the average time between false alarms (x-axis). This trade-off metric is illustrated in Figure 16, and its comparison to the performance of ROC curves (and other metrics that use the true positive rates as part of their calculations) against the adversary model we consider is illustrated in Figure 17.

Y-axis (Security). We consider a strong adversary model where the attacker knows all details about our anomaly detection test, and thus can remain undetected, even if we use *active* monitoring Given an anomaly detection threshold τ, we want to evaluate how much "damage" the attacker can do without raising an alarm.

The adversary wants to drive the system to the worst possible condition it can without being detected, where "worst" refers to *the maximum deviation of a signal from its true value that the attacker can obtain* (without raising an alarm, and given a fixed-period of time, otherwise given infinite time, the attacker might be able to grow this deviation without bound).

X-axis (Usability). While the y-axis of our proposed metric is completely different to ROC curves, the x-axis is similar, but instead of using the false-alarm rate, we use instead the expected time between false alarms $E[T_{fa}]$. This value has a couple of advantages over the false-alarm rate: (1) it

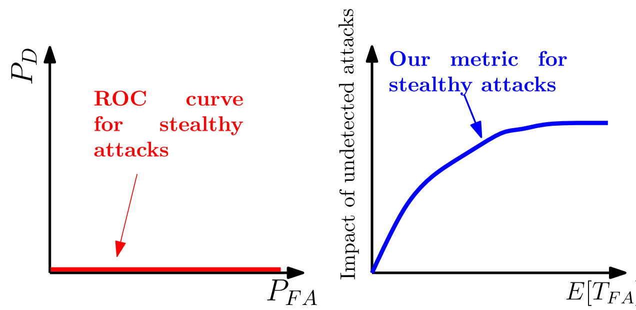

Fig. 17. Comparison of ROC curves with our proposed metric: ROC curves are not useful to measure the effectiveness of stealthy attacks.

addresses the deceptive nature of low false-alarm rates due to the base-rate fallacy [6], and (2) it addresses the problem that some anomaly detection statistics make a decision ("alarm" or "normal behavior") at non-constant time intervals.

7 RELATED WORK

There are several works that survey different aspects of security in cyber-physical systems from different domains. Several surveys [34, 43, 48, 59, 65, 99] focus their attention on security and privacy in smart grids and they explore different vulnerabilities in RTUs, IEDs, and smart meters that can be exploited by adversaries. Other surveys [2, 79, 87] analyze different types of attacks that can gain access to telemetry interfaces, software, and hardware in medical devices, with a special attention to implantable medical devices. Recently, there has been an increasing attention in manufacturing devices and the risks of including more sophisticated communication capabilities [81, 113, 121]. Approaches that do not focus on a specific domain but instead address security and privacy issues in a general CPS context include References [18, 32, 33, 39, 64, 72, 109]. Even though there is a wide number of surveys from different venues that address several aspects of security in CPS, we are the first ones that focus on physics-based anomaly detection and in proposing a unified taxonomy that include the vast amount of research in this field.

8 CONCLUSIONS

In this work, we introduced theoretical and practical contributions to the growing literature of physics-based attack detection in control systems. In particular, we provide a comprehensive taxonomy of related work and discuss general shortcomings we identified. We hope that by presenting multiple research papers in a unified way, we can motivate further discussion in this space and help other researchers develop the theoretical foundations, the language, and the tools to propose new attack models, or new metrics to address any limitations that our work may have.

We also showed that in the literature there is a not a unified methodology to evaluate and compare different detection mechanisms. In particular, we argued that using true positive rates assumes that attacks will be detected, but a sophisticated attacker can spoof deviations that follow relatively close to the "physics" of the system (launch undetected attacks) while still driving the system to a different state. We introduced the evaluation metric proposed in Reference [105] that can be used to compare different attack detection strategies by quantifying the maximum impact an attack can cause while remaining undetected. This is fundamentally different than any metric that uses true positives. Had we used ROC curves for our attacks, we would have obtained a flat line along the x-axis, because we have 0% detection rate. We believe this metric is a fundamental change to the way intrusion detection systems can be evaluated in the control systems space and we encourage the research community to use this metric or propose new metrics that allow the comparison of previous and current detection mechanisms even in the presence of stealthy attacks.

In all the systems we surveyed, the defender always assumed a fixed number of authenticated sensors and controllers, but an area that has been relatively unexplored in CPS is the concept of Sybil attacks, where attackers can create new sensors (and potentially controllers as well) to compete with honest sensor data. Sybil attacks can be particularly problematic in crowdsourced sensor data such as routing services like *Waze* [110]. As crowdsourcing becomes more prevalent in emerging CPS applications, Sybil attacks are a new type of attack that researchers will need to consider more in the future.

Future Work. There are many challenges for future research. For example, closing the gap between IT and control systems to design better time-series models while taking into account communications networks limitations and extending physics-based anomaly detection in other domains, such as manufacturing and transportation. Most of our experiments considered an attacker that wants to remain undetected, but in practice an attacker might sacrifice detection for achieving a desired malicious objective. An additional area of future research is how to respond to alerts.

REFERENCES

[1] Muhammad Qasim Ali and Ehab Al-Shaer. 2013. Configuration-based IDS for advanced metering infrastructure. In *Proceedings of the Conference on Computer & Communications Security (CCS'13)*. 451–462.
[2] Riham Al Tawy and Amr M. Youssef. 2016. Security tradeoffs in cyber physical systems: A case study survey on implantable medical devices. *IEEE Access* 4 (2016), 959–979.
[3] Saurabh Amin, Xavier Litrico, Shankar Sastry, and Alexandre M. Bayen. 2013. Cyber security of water SCADA systems–part I: analysis and experimentation of stealthy deception attacks. *IEEE Trans. Control Syst. Technol.* 21, 5 (2013), 1963–1970.
[4] Saurabh Amin, Xavier Litrico, S. Shankar Sastry, and Alexandre M. Bayen. 2013. Cyber security of water SCADA systems–part II: Attack detection using enhanced hydrodynamic models. *IEEE Trans. Control Syst. Technol.* 21, 5 (2013), 1679–1693.
[5] Karl Johan Åström and Peter Eykhoff. 1971. System identification: A survey. *Automatica* 7, 2 (1971), 123–162.
[6] Stefan Axelsson. 2000. The base-rate fallacy and the difficulty of intrusion detection. *ACM Trans. Info. Syst. Secur.* 3, 3 (2000), 186–205.
[7] Cheng-zong Bai and Vijay Gupta. 2014. On Kalman filtering in the presence of a compromised sensor: Fundamental performance bounds. In *Proceedings of American Control Conference*. 3029–3034.
[8] Cheng-zong Bai, Fabio Pasqualetti, and Vijay Gupta. 2015. Security in stochastic control systems: Fundamental limitations and performance bounds. In *Proceedings of American Control Conference*.
[9] Robin Berthier and William H. Sanders. 2011. Specification-based intrusion detection for advanced metering infrastructures. In *Proceedings of the Pacific Rim International Symposium on Dependable Computing (PRDC'11)*. IEEE, 184–193.
[10] Rakesh B. Bobba, Katherine M. Rogers, Qiyan Wang, Himanshu Khurana, Klara Nahrstedt, and Thomas J. Overbye. 2010. Detecting false data injection attacks on DC state estimation. In *Proceedings of Workshop on Secure Control Systems*, Vol. 2010.
[11] Paul Brooks. 2001. *EtherNet/IP: Industrial Protocol White Paper*. Technical Report. Rockwell Automation.
[12] Andrea Carcano, Alessio Coletta, Michele Guglielmi, Marcelo Masera, Igor Nai Fovino, and Alberto Trombetta. 2011. A multidimensional critical state analysis for detecting intrusions in SCADA systems. *IEEE Trans. Industr. Info.* 7, 2 (2011), 179–186.
[13] Alvaro A. Cardenas, Saurabh Amin, Zong-Syun Lin, Yu-Lun Huang, Chi-Yen Huang, and Shankar Sastry. 2011. Attacks against process control systems: Risk assessment, detection, and response. In *Proceedings of the ACM Symposium on Information, Computer and Communications Security*. 355–366.
[14] Alvaro A. Cárdenas, John S. Baras, and Karl Seamon. 2006. A framework for the evaluation of intrusion detection systems. In *Proceedings of Symposium on Security and Privacy*. IEEE, 15–pp.
[15] Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, Tadayoshi Kohno et al. 2011. Comprehensive experimental analyses of automotive attack surfaces. In *Proceedings of the USENIX Security Symposium*.

[16] Steven Cheung, Bruno Dutertre, Martin Fong, Ulf Lindqvist, Keith Skinner, and Alfonso Valdes. 2007. Using model-based intrusion detection for SCADA networks. In *Proceedings of the SCADA Security Scientific Symposium*, Vol. 46. 1–12.

[17] Michelle S. Chong, Masashi Wakaiki, and Joao P. Hespanha. 2015. Observability of linear systems under adversarial attacks. In *Proceedings of the American Control Conference (ACC'15).* IEEE, 2439–2444.

[18] Luis F. Cómbita, Jairo Giraldo, Alvaro A. Cárdenas, and Nicanor Quijano. 2015. Response and reconfiguration of cyber-physical control systems: A survey. In *Proceedings of the IEEE 2nd Colombian Conference on Automatic Control (CCAC'15).* IEEE, 1–6.

[19] Valentina Conotter, James F. O'Brien, and Hany Farid. 2012. Exposing digital forgeries in ballistic motion. *IEEE Trans. Info. Forensics Secur.* 7, 1 (2012), 283–296.

[20] Shuguang Cui, Zhu Han, Soummya Kar, Tung T. Kim, H. Vincent Poor, and Ali Tajer. 2012. Coordinated data-injection attack and detection in the smart grid: A detailed look at enriching detection solutions. *IEEE Signal Process. Mag.* 29, 5 (2012), 106–115.

[21] György Dán and Henrik Sandberg. 2010. Stealth attacks and protection schemes for state estimators in power systems. In *Proceedings of IEEE Smart Grid Commnunications Conference (SmartGridComm'10).*

[22] Katherine R. Davis, Kate L. Morrow, Rakesh Bobba, and Erich Heine. 2012. Power flow cyber attacks and perturbation-based defense. In *Proceedings of Conference on Smart Grid Communications (SmartGridComm'12).* IEEE, 342–347.

[23] Van Long Do, Lionel Fillatre, and Igor Nikiforov. 2014. A statistical method for detecting cyber/physical attacks on SCADA systems. In *Proceedings of Conference on Control Applications (CCA'14).* IEEE, 364–369.

[24] Emeka Eyisi and Xenofon Koutsoukos. 2014. Energy-based attack detection in networked control systems. In *Proceedings of the Conference on High Confidence Networked Systems (HiCoNS'14).* ACM, New York, NY, 115–124. Retrieved from DOI: https://doi.org/10.1145/2566468.2566472

[25] Nicolas Falliere, Liam O. Murchu, and Eric Chien. 2011. W32. stuxnet dossier. *White paper, Symantec Corp., Security Response.*

[26] Piotr Gaj, Jürgen Jasperneite, and Max Felser. 2013. Computer communication within industrial distributed environment–a survey. *IEEE Trans. Industr. Info.* 9, 1 (2013), 182–189.

[27] Ryan M. Gerdes, Chris Winstead, and Kevin Heaslip. 2013. CPS: An efficiency-motivated attack against autonomous vehicular transportation. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC'13).* ACM, 99–108.

[28] J. J. Gertler. 1988. Survey of model-based failure detection and isolation in complex plants. *IEEE Control Syst. Mag.* 8, 6 (1988), 3–11.

[29] Annarita Giani, Eilyan Bitar, Manuel Garcia, Miles McQueen, Pramod Khargonekar, and Kameshwar Poolla. 2011. Smart grid data integrity attacks: characterizations and countermeasures π. In *Proceedings of Conference on Smart Grid Communications (SmartGridComm'11).* IEEE, 232–237.

[30] Dina Hadžiosmanović, Lorenzo Simionato, Damiano Bolzoni, Emmanuele Zambon, and Sandro Etalle. 2012. N-gram against the machine: On the feasibility of the n-gram network analysis for binary protocols. In *Research in Attacks, Intrusions, and Defenses.* Springer, 354–373.

[31] Dina Hadžiosmanović, Robin Sommer, Emmanuele Zambon, and Pieter H. Hartel. 2014. Through the eye of the PLC: semantic security monitoring for industrial processes. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC'14).* ACM, 126–135.

[32] S. Han, M. Xie, H. H. Chen, and Y. Ling. 2014. Intrusion Detection in Cyber-Physical Systems: Techniques and Challenges. *IEEE Syst. J.* 8, 4 (Dec. 2014), 1052–1062.

[33] Hongmei He, Carsten Maple, Tim Watson, Ashutosh Tiwari, Jörn Mehnen, Yaochu Jin, and Bogdan Gabrys. 2016. The security challenges in the IoT enabled cyber-physical systems and opportunities for evolutionary computing & other computational intelligence. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'16).* IEEE, 1015–1021.

[34] Haibo He and Jun Yan. 2016. Cyber-physical attacks and defences in the smart grid: a survey. *IET Cyber-Phys. Syst.: Theory Appl.* 1, 1 (2016), 13–27.

[35] Xiali Hei, Xiaojiang Du, Shan Lin, and Insup Lee. 2013. PIPAC: Patient infusion pattern based access control scheme for wireless insulin pump system. In *Proceedings of INFOCOM.* IEEE, 3030–3038.

[36] Nathan Henry, Nathanael Paul, and Nicole McFarlane. 2013. Using bowel sounds to create a forensically aware insulin pump system. In *Proceedings of Workshop on Health Information Technologies.*

[37] Baik Hoh, Marco Gruteser, Ryan Herring, Jeff Ban, Daniel Work, Juan-Carlos Herrera, Alexandre M. Bayen, Murali Annavaram, and Quinn Jacobson. 2008. Virtual trip lines for distributed privacy-preserving traffic monitoring. In *Proceedings of the Conference on Mobile Systems, Applications, and Services.* ACM, 15–28.

[38] Fangyuan Hou, Zhonghua Pang, Yuguo Zhou, and Dehui Sun. 2015. False data injection attacks for a class of output tracking control systems. In *Proceedings of Chinese Control and Decision Conference.* 3319–3323.

[39] J. How. 2015. Cyberphysical security in networked control systems [about this issue]. *IEEE Control Syst.* 35, 1 (Feb. 2015), 8–12. Retrieved from DOI: https://doi.org/10.1109/MCS.2014.2364693

[40] Abdulmalik Humayed, Jingqiang Lin, Fengjun Li, and Bo Luo. 2017. Cyber-physical systems security—A survey. *arXiv preprint arXiv:1701.04525* (2017).

[41] Inseok Hwang, Sungwan Kim, Youdan Kim, and Chze Eng Seah. 2010. A survey of fault detection, isolation, and reconfiguration methods. *IEEE Trans. Control Syst. Technol.* 18, 3 (2010), 636–653.

[42] Rob Millerb Ishtiaq Roufa, Hossen Mustafaa, Sangho Ohb Travis Taylora, Wenyuan Xua, Marco Gruteserb, Wade Trappeb, and Ivan Seskarb. 2010. Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study. In *Proceedings of USENIX Security Symposium.* 11–13.

[43] Marek Jawurek, Florian Kerschbaum, and George Danezis. 2012. *Privacy Technologies for Smart Grids—A Survey of Options.* Technical Report MSR-TR-2012-119. Retrieved from http://research.microsoft.com/apps/pubs/default.aspx?id=178055.

[44] K. H. Johansson. 2000. The quadruple-tank process: A multivariable laboratory process with an adjustable zero. *IEEE Trans. Control Syst. Technol.* 8, 3 (May 2000), 456–465. Retrieved from DOI: https://doi.org/10.1109/87.845876

[45] Andrew J. Kerns, Daniel P. Shepard, Jahshan A. Bhatti, and Todd E. Humphreys. 2014. Unmanned aircraft capture and control via GPS spoofing. *J. Field Robot.* 31, 4 (2014), 617–636.

[46] Tùng T. Kim and H. Vincent Poor. 2011. Strategic protection against data injection attacks on power grids. *IEEE Trans. Smart Grid* 2, 2 (2011), 326–333.

[47] Istvan Kiss, Bela Genge, and Piroska Haller. 2015. A clustering-based approach to detect cyber attacks in process control systems. In *Proceedings of Conference on Industrial Informatics (INDIN'15).* IEEE, 142–148.

[48] Nikos Komninos, Eleni Philippou, and Andreas Pitsillides. 2014. Survey in smart grid and smart home security: Issues, challenges and countermeasures. *IEEE Commun. Surveys Tutor.* 16, 4 (2014), 1933–1954.

[49] Karl Koscher, Alexei Czeskis, Franziska Roesner, Shwetak Patel, Tadayoshi Kohno, Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham et al. 2010. Experimental security analysis of a modern automobile. In *Proceedings of the IEEE Symposium on Security and Privacy (SP'10).* IEEE, 447–462.

[50] Oliver Kosut, Liyan Jia, Robert Thomas, and Lang Tong. 2010. Malicious Data Attacks on Smart Grid State Estimation: Attack Strategies and Countermeasures. In *Proceedings of IEEE Smart Grid Commnunications Conference (SmartGridComm'10).*

[51] Georgia Koutsandria, Vishak Muthukumar, Masood Parvania, Sean Peisert, Chuck McParland, and Anna Scaglione. 2014. A Hybrid Network IDS for Protective Digital Relays in the Power Transmission Grid. In *Proceedings of Conference on Smart Grid Communications (SmartGridComm'14).*

[52] Marina Krotofil, Jason Larsen, and Dieter Gollmann. 2015. The process matters: Ensuring data veracity in cyber-physical systems. In *Proceedings of the ACM Symposium on Information, Computer and Communications Security (ACSAC'15).* ACM, 133–144.

[53] Cheolhyeon Kwon, Weiyi Liu, and Inseok Hwang. 2013. Security analysis for cyber-physical systems against stealthy deception attacks. In *Proceedings of American Control Conference.* 3344–3349.

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

[55] Michael LeMay and Carl A. Gunter. 2012. Cumulative attestation kernels for embedded systems. *IEEE Trans. Smart Grid* 3, 2 (2012), 744–760.

[56] Jingwen Liang, Oliver Kosut, and Lalitha Sankar. 2014. Cyber attacks on AC state estimation: Unobservability and physical consequences. In *Proceedings of PES General Meeting.* 1–5. Retrieved from DOI: https://doi.org/10.1109/PESGM.2014.6939486

[57] Hui Lin, Adam Slagell, Catello Di Martino, Zbigniew Kalbarczyk, and Ravishankar K. Iyer. 2013. Adapting Bro into SCADA: Building a specification-based intrusion detection system for the DNP3 protocol. In *Proceedings of the 8th Annual Cyber Security and Information Intelligence Research Workshop.* ACM, 5.

[58] Hui Lin, Adam Slagell, Zbigniew Kalbarczyk, Peter W. Sauer, and Ravishankar K. Iyer. 2013. Semantic security analysis of SCADA networks to detect malicious control commands in power grids. In *Proceedings of the ACM Workshop on Smart Energy Grid Security.* ACM, 29–34.

[59] Jing Liu, Yang Xiao, Shuhui Li, Wei Liang, and C. L. Philip Chen. 2012. Cyber security and privacy issues in smart grids. *IEEE Commun. Surveys Tutor.* 14, 4 (2012), 981–997.

[60] Yao Liu, Peng Ning, and Michael K. Reiter. 2009. False data injection attacks against state estimation in electric power grids. In *Proceedings of the conference on Computer and communications security (CCS'09).* ACM, 21–32.

[61] Yao Liu, Peng Ning, and Michael K. Reiter. 2011. False data injection attacks against state estimation in electric power grids. *ACM Trans. Info. Syst. Secur.* 14, 1 (2011), 13.

[62] L. Ljung. 1996. *The Control Handbook.* CRC Press, Chapter System Identification, 1033–1054.

[63] Lennart Ljung (Ed.). 1999. *System Identification* (2nd Ed.)*: Theory for the User.* Prentice Hall PTR, Upper Saddle River, NJ.

[64] Yuriy Zacchia Lun, Alessandro D'Innocenzo, Ivano Malavolta, and Maria Domenica Di Benedetto. 2016. Cyber-physical systems security: A systematic mapping study. *arXiv preprint arXiv:1605.09641* (2016).

[65] M. H. Cintuglu and O. A. Mohammed and K. Akkaya and A. S. Uluagac. 2017. A survey on smart grid cyber-physical system testbeds. *IEEE Commun. Surveys Tutor.* 19, 1 (2017), 446–464.

[66] Daisuke Mashima and Alvaro Cárdenas. 2012. Evaluating electricity theft detectors in smart grid networks. In *Research in Attacks, Intrusions, and Defenses.* Springer, 210–229.

[67] Stephen McLaughlin. 2013. CPS: Stateful policy enforcement for control system device usage. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC'13).* ACM, New York, NY, USA, 109–118.

[68] Stephen McLaughlin and Patrick McDaniel. 2012. SABOT: Specification-based payload generation for programmable logic controllers. In *Proceedings of the Conference on Computer and Communications Security (CCS'12).* ACM, 439–449.

[69] Stephen McLaughlin, Saman Zonouz, Devin Pohly, and Patrick McDaniel. 2014. A trusted safety verifier for process controller code. In *Proceedings of the ISOC Network and Distributed Systems Security Symposium (NDSS'14).*

[70] Fei Miao, Quanyan Zhu, Miroslav Pajic, and George J. Pappas. 2014. Coding sensor outputs for injection attacks detection. In *Proceedings of Conference on Decision and Control.* 5776–5781.

[71] Shaunak Mishra, Yasser Shoukry, Nikhil Karamchandani, Suhas N. Diggavi, and Paulo Tabuada. 2017. Secure state estimation against sensor attacks in the presence of noise. *IEEE Trans. Control Netw. Syst.* 4, 1 (2017), 49–59.

[72] Robert Mitchell and Ing-Ray Chen. 2014. A survey of intrusion detection techniques for cyber-physical systems. *ACM Comput. Surv.* 46, 4, Article 55 (Mar. 2014), 29 pages.

[73] Sayan Mitra, Tichakorn Wongpiromsarn, and Richard M. Murray. 2013. Verifying cyber-physical interactions in safety-critical systems. *IEEE Secur. Privacy* 11, 4 (2013), 28–37.

[74] Yilin Mo and Bruno Sinopoli. 2009. Secure control against replay attacks. In *Proceedings of the Annual Allerton Conference on Communication, Control, and Computing (Allerton'09).* IEEE, 911–918.

[75] Yilin Mo, Sean Weerakkody, and Bruno Sinopoli. 2015. Physical authentication of control systems: Designing watermarked control inputs to detect counterfeit sensor outputs. *IEEE Control Syst.* 35, 1 (2015), 93–109.

[76] Y. L. Mo, R. Chabukswar, and B. Sinopoli. 2014. Detecting integrity attacks on SCADA systems. *IEEE Trans. Control Syst. Technol.* 22, 4 (2014), 1396–1407.

[77] Kate L. Morrow, Erich Heine, Katherine M. Rogers, Rakesh B. Bobba, and Thomas J. Overbye. 2012. Topology perturbation for detecting malicious data injection. In *Proceedings of Conference on System Science (HICSS'12).* IEEE, 2104–2113.

[78] C. Murguia and J. Ruths. 2016. Characterization of a CUSUM model-based sensor attack detector. In *Proceedings of the IEEE 55th Conference on Decision and Control (CDC'16).* 1303–1309.

[79] O. Kocabas, T. Soyata, and M. K. Aktas. 2016. Emerging security mechanisms for medical cyber physical systems. *IEEE/ACM Trans. Comput. Biol. Bioinformat.* 13, 3 (May 2016), 401–416.

[80] ODVA: The CIP Networks Library, Volume 2, "EtherNet/IP Adaptation of CIP", Edition 1.11, April 2011. Accessed on July 15, 2017.

[81] Yao Pan, Jules White, Douglas C. Schmidt, Ahmad Elhabashy, Logan Sturm, Jaime Camelio, and Christopher Williams. 2017. Taxonomies for reasoning about cyber-physical attacks in IoT-based manufacturing systems. *Int. J. Interact. Multimedia Artific. Intel.* 4, Special Issue on Advances and Applications in the Internet of Things and Cloud Computing (2017).

[82] M. Parvania, G. Koutsandria, V. Muthukumary, S. Peisert, C. McParland, and A. Scaglione. 2014. Hybrid control network intrusion detection systems for automated power distribution systems. In *Proceedings of Conference on Dependable Systems and Networks (DSN'14).* 774–779.

[83] F. Pasqualetti, F. Dorfler, and F. Bullo. 2013. Attack detection and identification in cyber-physical systems. *IEEE Trans. Auto. Control* 58, 11 (Nov. 2013), 2715–2729.

[84] Mohammad Ashiqur Rahman, Ehab Al-Shaer, Md Rahman et al. 2013. A formal model for verifying stealthy attacks on state estimation in power grids. In *Proceedings of Conference on Smart Grid Communications (SmartGridComm'13).* IEEE, 414–419.

[85] Ishtiaq Rouf, Hossen Mustafa, Miao Xu, Wenyuan Xu, Rob Miller, and Marco Gruteser. 2012. Neighborhood watch: Security and privacy analysis of automatic meter reading systems. In *Proceedings of the conference on Computer and communications security (CCS'12).* ACM, 462–473.

[86] Michael Rushanan, Aviel D. Rubin, Denis Foo Kune, and Colleen M. Swanson. 2014. SoK: Security and privacy in implantable medical devices and body area networks. In *Proceedings of Symposium on Security and Privacy (S&P'14).* IEEE.

[87] Michael Rushanan, Aviel D. Rubin, Denis Foo Kune, and Colleen M. Swanson. 2014. SoK: Security and privacy in implantable medical devices and body area networks. In *Proceedings of the IEEE Symposium on Security and Privacy (SP'14).* IEEE, 524–539.

[88] Imran Sajjad, Daniel D. Dunn, Rajnikant Sharma, and Ryan Gerdes. 2015. Attack mitigation in adversarial platooning using detection-based sliding mode control. In *Proceedings of the ACM Workshop on Cyber-Physical Systems-Security and/or Privacy (CPS-SPC'15).* ACM, New York, NY, 43–53.

[89] Imran Sajjad, Daniel D. Dunn, Rajnikant Sharma, and Ryan Gerdes. 2015. Attack mitigation in adversarial platooning using detection-based sliding mode control. In *Proceedings of the Workshop on Cyber-Physical Systems-Security and/or Privacy (CPS-SPC'15).* ACM, New York, NY, 43–53. Retrieved from DOI: https://doi.org/10.1145/2808705.2808713

[90] Henrik Sandberg, André Teixeira, and Karl H. Johansson. 2010. On security indices for state estimators in power networks. In *Proceedings of the Workshop on Secure Control Systems.*

[91] Reza Shokri, George Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux. 2011. Quantifying location privacy. In *Proceedings of Symposium on Security and Privacy (S&P'11).* IEEE, 247–262.

[92] Yasser Shoukry, Michelle Chong, Masashi Wakaiki, Pierluigi Nuzzo, Alberto L Sangiovanni-Vincentelli, Sanjit A. Seshia, Joao P. Hespanha, and Paulo Tabuada. 2016. SMT-based observer design for cyber-physical systems under sensor attacks. In *Proceedings fo the ACM/IEEE 7th International Conference on Cyber-Physical Systems (ICCPS'16).* IEEE, 1–10.

[93] Yasser Shoukry, Paul Martin, Yair Yona, Suhas Diggavi, and Mani Srivastava. 2015. PyCRA: Physical challenge-response authentication for active sensors under spoofing attacks. In *Proceedings of the Conference on Computer and Communications Security (CCS'15).* ACM, New York, NY, 1004–1015.

[94] Roy Smith. 2011. A decoupled feedback structure for covertly appropriating networked control systems. In *Proceedings of World Congress*, Vol. 18. 90–95.

[95] R. S. Smith. 2015. Covert misappropriation of networked control systems: Presenting a feedback structure. *IEEE Control Syst.* 35, 1 (Feb. 2015), 82–92. DOI: https://doi.org/10.1109/MCS.2014.2364723

[96] Eduardo D. Sontag. 1998. *Mathematical Control Theory: Deterministic Finite Dimensional Systems.* Vol. 6. Springer.

[97] Siddharth Sridhar and Manimaran Govindarasu. 2014. Model-based attack detection and mitigation for automatic generation control. *IEEE Trans. Smart Grid* 5, 2 (2014), 580–591.

[98] Rui Tan, Varun Badrinath Krishna, David K. Y. Yau, and Zbigniew Kalbarczyk. 2013. Impact of integrity attacks on real-time pricing in smart grids. In *Proceedings of the Conference on Computer & Communications Security (CCS'13).* ACM, 439–450.

[99] S. Tan, D. De, W. Z. Song, J. Yang, and S. K. Das. 2017. Survey of security advances in smart grid: A data driven approach. *IEEE Commun. Surveys Tutor.* 19, 1 (Firstquarter 2017), 397–422.

[100] André Teixeira, Saurabh Amin, Henrik Sandberg, Karl Henrik Johansson, and Shankar S. Sastry. 2010. Cyber security analysis of state estimators in electric power systems. In *Proceedings of Conference on Decision and Control (CDC'10).* IEEE, 5991–5998.

[101] André Teixeira, György Dán, Henrik Sandberg, and Karl Henrik Johansson. 2011. A cyber security study of a SCADA energy management system: Stealthy deception attacks on the state estimator. In *Proceedings of World Congress*, Vol. 18. 11271–11277.

[102] André Teixeira, Daniel Pérez, Henrik Sandberg, and Karl Henrik Johansson. 2012. Attack models and scenarios for networked control systems. In *Proceedings of the conference on High Confidence Networked Systems.* ACM, 55–64.

[103] André Teixeira, Iman Shames, Henrik Sandberg, and Karl Henrik Johansson. 2012. Revealing stealthy attacks in control systems. In *Proceedings of the Annual Allerton Conference on Communication, Control, and Computing (Allerton'12).* IEEE, 1806–1813.

[104] David Urbina, Jairo Giraldo, Nils Ole Tippenhauer, and Alvaro Cárdenas. 2016. Attacking fieldbus communications in ICS: Applications to the SWaT testbed. In *Proceedings of Singapore Cyber Security R&D Conference (SG-CRC'16)*, Vol. 14. 75–89.

[105] David I. Urbina, Jairo A. Giraldo, Alvaro A. Cardenas, Nils Ole Tippenhauer, Junia Valente, Mustafa Faisal, Justin Ruths, Richard Candell, and Henrik Sandberg. 2016. Limiting the impact of stealthy attacks on industrial control systems. In *Proceedings of the Conference on Computer and Communications Security (CCS'16).* ACM, 1092–1105.

[106] Junia Valente and Alvaro A. Cardenas. 2015. Using visual challenges to verify the integrity of security cameras. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC'15).* ACM.

[107] Junia Valente and Alvaro A. Cardenas. 2017. Remote proofs of video freshness for public spaces. In *Proceedings of the 2017 Workshop on Cyber-Physical Systems Security and PrivaCy (CPS'17).* ACM, New York, NY, 111–122.

[108] Ognjen Vuković and György Dán. 2013. On the security of distributed power system state estimation under targeted attacks. In *Proceedings of the Annual ACM Symposium on Applied Computing (ACSAC'13).* ACM, 666–672.

[109] Dong Wang, Zidong Wang, Bo Shen, Fuad E. Alsaadi, and Tasawar Hayat. 2016. Recent advances on filtering and control for cyber-physical systems under security and resource constraints. *J. Franklin Inst.* 353, 11 (2016), 2451–2466.

[110] Gang Wang, Bolun Wang, Tianyi Wang, Ana Nika, Haitao Zheng, and Ben Y. Zhao. 2016. Defending against Sybil devices in crowdsourced mapping services. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services.* ACM, 179–191.

[111] Yong Wang, Zhaoyan Xu, Jialong Zhang, Lei Xu, Haopei Wang, and Guofei Gu. 2014. SRID: State relation based intrusion detection for false data injection attacks in SCADA. In *Proceedings of European Symposium on Research in Computer Security (ESORICS'14)*. Springer, 401–418.

[112] Greg Welch and Gary Bishop. 1995. An introduction to the Kalman filter.

[113] Lee J. Wells, Jaime A. Camelio, Christopher B. Williams, and Jules White. 2014. Cyber-physical security challenges in manufacturing systems. *Manufact. Lett.* 2, 2 (2014), 74–77. https://doi.org/10.1016/j.mfglet.2014.01.005

[114] Theodore J. Williams. 1994. The purdue enterprise reference architecture. *Comput. Industry* 24, 2 (1994), 141–158.

[115] Alan S. Willsky. 1976. A survey of design methods for failure detection in dynamic systems. *Automatica* 12, 6 (1976), 601–611.

[116] Wireshark Network Protocol Analyzer. 2016. Retrieved from https://www.wireshark.org/.

[117] xnetfilterq. 2015. Python bindings for libnetfilter_queue. Retrieved from https://github.com/fqrouter/python-netfilterqueue.

[118] xpython. 2015. Python language. Version 2.7.10. Retrieved from https://docs.python.org/2/.

[119] xscapy. 2015. Scapy packet manipulation program. Version 2.3.1. Retrieved from http://www.secdev.org/projects/scapy/doc/.

[120] S. Z. Yong, M. Q. Foo, and E. Frazzoli. 2016. Robust and resilient estimation for cyber-physical systems under adversarial attacks. In *Proceedings of the American Control Conference (ACC'16)*. 308–315.

[121] Steven Eric Zeltmann, Nikhil Gupta, Nektarios Georgios Tsoutsos, Michail Maniatakos, Jeyavijayan Rajendran, and Ramesh Karri. 2016. Manufacturing and security challenges in 3D printing. *J. Miner. Metals Mater.* 68, 7 (2016), 1872–1881.

Received May 2017; revised March 2018; accepted March 2018